IBM Cloud Pak for AIOps 4.2.0 - PDF 1



© Copyright IBM Corp. 2025.
US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Tables of Contents

IBM Cloud Pak for AIOps 4.2.0 1

Release notes 1
- What's new 2
  - Known issues and limitations 4
  - Deprecated and removed functions 28

Overview 30
- Architecture 37
- Regulatory compliance 40
  - Federal Information Processing Standards (FIPS) 41
    - Enabling FIPS support for IBM Cloud Pak for AIOps 42
    - Enabling FIPS support for Infrastructure Automation 44
    - Enabling FIPS support for a Netcool connection 46
  - Audit logging 48
    - Logged actions and data 50
    - Audit log messages 53
    - Enabling Audit for Infrastructure Automation {{site.data.keyword.infra_mgmt_manageservices}} 61
  - Security and Privacy by Design (SPbD) 62
  - GDPR readiness 62
  - Data retention 66

Getting started 67
- Roles and archetypes 68
- IBM Cloud Pak for AIOps Workflows 69
  - Installing the IBM Cloud Pak 69
  - Guided tours 69
  - Defining data connections and integrations 69
  - Training AI models 70
  - Building resources and applications 70
  - Building topologies 69
  - Resolving incidents and alerts 70
  - Automating processes 69
  - Administering your cluster 71
  - Administering the Cloud Pak 71

Installing 71
- Planning 71
  - Supported OpenShift versions 72
  - Supported cloud platforms 73
  - Storage requirements 74
  - Hardware requirements 78
  - IBM Cloud Pak for AIOps ports 83
  - Operator installation mode 85
  - High availability considerations 86
    - Multi-zone HADR 86
  - Security considerations 89
    - Verifying images 90
    - Downloading or listing images 92
  - Licensing 93
    - License Service overview 94
    - Hyperthreading 96
    - Audit snapshot 97
    - Reported metrics 100
    - Enabling optional features 101
    - Viewing license usage on the Administration panel 106

Retrieving license usage data 107
Verifying completeness of data and troubleshooting 107
Using custom certificates 110
Tracking license usage in a multicluster environment 111
Tracking license usage with License Service Reporter 112
Deploying License Service Reporter 114
Configuring data sources 117
Validating the deployment 121
Viewing license usage on the Licensing dashboard 122
Manually tracking license usage 125
Passport Advantage part numbers 126
IBM Cloud Pak for AIOps 128
Starter installation 129
Production installation 136
Online (console) 136
Online (CLI) 146
Offline 155
Bastion host 157
Portable device 171
Installing on cloud platforms 187
Installing on ARO 188
Installing on GCP 193
Installing on ROKS 195
Installing on ROSA 196
Infrastructure Automation 201
Installing stand-alone Infrastructure Automation 201
Online (CLI) 202
Offline 209
Installing appliances 210
Installing on AWS 211
Installing on VMware vSphere 219
Installing on Red Hat OpenStack Platform 227
Installing on Red Hat Virtualization 237
Installing for use with IBM Cloud Pak for AIOps 245
Online (CLI) 245
Offline 257
Bastion host 259
Portable compute device 266
Portable storage device 274
Post installation for use with IBM Cloud Pak for AIOps 283
Deploying Infrastructure management operand 283
Deploying Infrastructure management as a containerized deployment (podified) 283
Deploying Infrastructure management as a virtual machine appliance 285
OIDC Configuration Steps for Infrastructure Management Appliance only 286

Upgrading 292
IBM Cloud Pak for AIOps 293
Online (console) 294
Online (CLI) 300
Offline 307
Configuring automatic patch upgrades 314
Infrastructure Automation 316
Online (CLI) 317
Offline 319
Upgrading PostgreSQL for a podified Infrastructure management installation 321
Migrating to Infrastructure management appliance 323
Upgrading Infrastructure management appliances 327

Upgrade an Infrastructure management appliance from 3.6.x 328
Upgrade an Infrastructure management VMDB appliance 329
Upgrade an Infrastructure management Non-VMDB appliance 332
Upgrade a dedicated database appliance 333
Migrate from an appliance-based to a Kubernetes-based Infrastructure Management installation 335

Uninstalling 340
IBM Cloud Pak for AIOps 341
Infrastructure Automation 342

Defining data connections and integrations 344
Netcool integrations 344
Data connections 345
Data and tool connections UI 346
Connection types 347
Outgoing connections 348
Microsoft Teams connection 348
Configuring Microsoft Teams application for connection 351
Slack connection 353
Configuring Slack for connection 355
ServiceNow App connection 358
Creating a Change Risk Assessment 365
Incoming connections 366
Event and metric connections 367
Ansible connections 367
Appdynamics connections 368
AWS CloudWatch connections 371
Custom connections 374
Creating custom connections using the Connector SDK 382
Dynatrace connections 382
ELK connections 385
Generating API Key for ELK connections 389
EFK connections 390
Falcon LogScale connections 396
Generic webhook connection 401
Example: IBM SevOne NMS mappings 403
Example: Prometheus AlertManager mappings 407
IBM Tivoli Netcool/Impact connection 411
Configuring a connection in IBM Tivoli Netcool/Impact 414
Instana connections 417
Kafka connections 422
LogDNA connections 426
Metric Manager connections 431
IBM Tivoli Netcool/OMNIbus connections 433
Configuring launch-in-context from IBM Tivoli Netcool/OMNIbus 437
Adding incidents and alerts to the DASH navigation menu 438
Creating tools for launch-in-context from the Event Viewer 439
Creating a TLS-enabled connection to an OpenShift ObjectServer 440
Filter clause 441
New Relic connections 442
Observer connections 445
PagerDuty connections 445
ServiceNow connection 447
Mapping similar ticket data from ServiceNow 457
Splunk connections 459
SSH connections 464

Zabbix connections 466
Delay configurations in data connections 470
Mapping data 470
Log data 471
Event data 479
Performance considerations for metric data collection 483
Loading data for AI training 484
Loading data from domain-specific log sources 485
Deduplication 486
HTTP headers for connector credentials 487
Remote deployment of connectors 487
Observer jobs 488
ALM observer jobs 493
Ansible AWX observer jobs 495
AppDynamics observer jobs 497
AWS observer jobs 498
Azure observer jobs 499
Big Cloud Fabric observer jobs 501
BigFix Inventory observer jobs 503
Ciena Blue Planet observer jobs 505
Cisco ACI observer jobs 506
Contrail observer jobs 509
Datadog observer jobs 512
DNS observer jobs 513
Docker observer jobs 515
Dynatrace observer jobs 517
File observer jobs 520
GoogleCloud observer jobs 523
HP NFVD observer jobs 525
IBM Cloud observer jobs 531
Jenkins observer jobs 532
Configuring Jenkins observer job plug-in 533
More Jenkins plug-in customization tasks 536
Customizing the Jenkins topologies 536
Creating Jenkins topology templates 538
Juniper CSO observer jobs 539
Kubernetes observer jobs 540
Network Manager observer jobs 548
New Relic observer jobs 549
OpenStack observer jobs 551
Rancher observer jobs 555
REST observer jobs 556
SDC ONAP observer jobs 562
ServiceNow observer jobs 565
SevOne observer jobs 569
Tivoli Application Dependency Discovery Manager observer jobs 570
Viptela observer jobs 572
VMware NSX observer jobs 574
VMware vCenter observer jobs 575
Zabbix observer jobs 577
Configuring observer job security 578
Probes and probe integrations 581
Connecting with on-premises probes and uni-directional gateway 581
Configuring the Cloud system 582
Configuring the on-premises system 584
IBM SevOne NPM probe integration 585
IBM SevOne NPM probe integration 586
IBM Turbonomic probe integration 589
Datatog probe integration 594

Probe Common Tasks 598

Viewing AIOps insights 599
AIOps insights UI 599
AIOps insights Scopes and Filter 606

Training AI models 608
About AI models 608
Prerequisite AI models 610
Preparing the AI environment 611
Adjusting model training resources 612
Configuring alert expiry time 612
Improving data streaming capacity 613
About Elasticsearch indices and shards 615
Determining your current shard consumption 615
Increasing the number of Elasticsearch shards 618
Changing the number of shards per logtrain index 619
Managing embedded JSON 620
Scaling log anomaly detection pods 621
Configuring log formats for WebSphere log anomaly detection 622
About AI algorithms 622
About change risk 623
About event grouping 624
About log anomaly detection algorithms 625
About natural language log anomaly detection 626
Workflow 626
Example 630
About statistical baseline log anomaly detection 636
Workflow 636
Example 638
About metric anomaly detection 643
About probable cause ranking 644
About similar tickets 646
Language support 647
Preparing AI algorithms 648
About algorithm types 648
About pre-trained algorithms 649
Configuring trainable AI algorithms 650
Change risk 650
Metric anomaly detection 652
Natural language log anomaly detection 653
Similar tickets 658
Temporal grouping 660
Managing AI algorithms 661
Generating AI models 662
Displaying log anomaly detection resources and models 664
Data quality messages 664
Managing unsupported languages 667
Determining which algorithm generated a log anomaly 668
Deploying models 669
Editing algorithm configurations 670
Deleting algorithm configurations 671
Managing AI model versions 672
Changing the deployed version 672
Deleting a version 673
Draft and archived policies 675
Displaying data assets 676

Managing applications (services) and viewing topologies 677

Resource management: Searching and filtering 678
The Resource management page 680
Defining and editing applications (services)) 683
Managing topologies 686
Rendering a topology 686
Building a custom topology 689
Viewing a topology 690
Viewing topology history 701
Viewing topologies with URLs 703
Viewing topologies with API 706
Topology query syntax 708
Template types 711
Viewing resource group templates 712
Building resource group templates 713
Incidents 714
Defining business criticality 716
Running data administration routines 717
Configuring topology user preferences 718
Configuring topology elements 720
Configuring advanced settings 721
Configuring topology tools 723
Splunk integration example 729
Configuring resource types 730
Configuring relationship types 732
Configuring custom icons 734
Configuring rules 736

Resolving incidents and alerts 739
About events, alerts and incidents 739
Incidents 741
Incident overview 742
Searching and filtering incidents 746
Managing multiple incidents 746
Alerts 746
Icons used in the Alert Viewer 748
Alert details 749
Working with alerts 751
Searching and filtering alerts 757
Creating quick filters 759
Setting user preferences for the Alert Viewer 760
About alert groups 761
Displaying an alert group 761
Displaying alert topology 762
Displaying alert seasonality 762
Displaying temporal correlation 764
Log anomaly alerts 766
High risk alerts from change risk 767
Viewing metric anomaly details 768
Metric anomaly forecasts 772
Viewing multiple metrics on the metric anomaly chart 772
Using Metric search to create a timeseries chart 775
Displaying analytics details for an alert group 776
Viewing incidents that are displayed in outgoing connections 778
Event mapping and alert severity 779
ChatOps insights 781
Managing Incidents in ChatOps 782
Probable cause in ChatOps 784
Incident Resources 785
Recommended runbooks 786

Recommended actions 786
Viewing alerts in ChatOps 788
Change risk in ChatOps 792

Automating processes 796
Policies 796
Searching and filtering policies 799
Policy details 800
Seasonality policy details 800
Temporal policy details 802
Default incident creation policies 803
Incident Creation based on Persistence and Golden Signal 804
Creating policies 805
Suppress alerts 806
Suppress seasonal alerts 808
Group alerts based on scope 809
Assign a runbook to alerts 810
Promote alerts to an incident 812
Invoke an Impact policy 814
Supported Anomaly insights properties 817
Policy editor 820
Best practices for condition sets 825
Refreshing the policies table 825
Managing multiple policies 826
Renaming policies 826
Archiving policies 827
Runbooks 827
Runbooks and Actions 828
Runbooks Library 828
Run your first runbook 829
Load and reset example runbooks 830
Create a runbook 832
Edit a runbook 833
Runbook elements and editing actions 834
Adding actions 835
Adding nested runbooks 836
Adding parameters 837
Parameter types 838
Adding a GOTO element 840
Configuring runbook creation 840
Configuring runbook approval 841
Runbook approval process 841
Runbook versions 842
Save and publish a runbook 843
Filtering runbooks 843
Assigning runbooks to groups 844
Export and import runbooks 844
Delete a runbook 846
Activities 847
Monitor runbook history 847
Actions 848
Managing actions 848
Creating an action 849
Creating SSH actions 850
Single and multi target actions 853
Output format of multi target actions 854
Comparison of single and multi target actions 854

Creating HTTP actions 855
Creating Client-side actions 857
Ansible actions 858
Job Template parameters and surveys 858
Harvesting Ansible actions 859
Ansible action interactions 859
Predefined actions 859
Action versions 860
Save and publish an action 861
Testing an action 861
Running an action 861
Activities 862

Administering the Cloud Pak 863
CLI tools 863
Installing the IBM Cloud Pak CLI (cloudctl) 863
Installing the Kubernetes CLI (kubectl) 864
Using a custom certificate 865
Secret rotation 866
Issue Resolution Core operator secret usage 866
Issue Resolution Lifecycle operator secret usage 877
UI console 879
Administration panel 880
Accessing the console 882
Supported browsers 883
Configuring the console logo 883
Customizing the platform UI 884
Enabling and disabling tours 885
Custom cards API 885
Customizing the console branding 895
Customizing the home page 896
Setting your preferences 898
Managing user access control 899
Roles and permissions 900
Adding users 910
Managing roles 912
Managing user groups 913
Configuring LDAP servers and authentication 915
Multiple LDAP domains 916
Authentication types 917
Delegating authentication to OpenShift 919
Configuring foundational services IAM for single sign-on 921
Configuring single sign-on 925
Configuring LDAP connections 929
Changing LDAP cache settings 933
Changing LDAP search settings 934
Changing LDAP search cache variable values 937
Updating SCIM LDAP attributes mapping 939
Changing Logjam configuration property 947
Managing namespaces 947
Creating a namespace 948
Removing a namespace 949
Namespace limits 949
Managing Kubernetes Secret passwords 951
Changing the cluster administrator credentials 952
Managing certificates 955
Installing certificates 956
Certificate management (operator) 957

Accessing the foundational services endpoint certificate 957
Replacing foundational services endpoint certificates 957
Refreshing foundational services internal certificates 962
Creating your own self-signed and CA Issuers 968
Creating Certificate manager (cert-manager) certificates 969
Customizing cert-manager certificates 971
Viewing cert-manager resources 973
Refreshing cert-manager certificates 974
Using Vault to issue certificates 976
Adding certificates by using the ECDSA algorithm for encryption 979
Certificate Manager role-based access control (RBAC) support 981
Secure Tunnels 982
Installing Secure Tunnel 985
Installing the Connector 990
Upgrading Secure Tunnel 995
Creating Secure Tunnel Connections 997
Viewing Secure Tunnel 1010
Modifying Secure Tunnel 1012
Auditing Secure Tunnel 1013
Restarting the Cloud Pak for AIOps cluster 1016
Backing up and restoring Cloud Pak for AIOps 1021
Installing the backup and restore tools 1025
Backing up Cloud Pak for AIOps 1029
Restoring Cloud Pak for AIOps 1034
Upgrading IBM Cloud Pak for AIOps backup and restore artifacts 1049
Administering data storage 1051
Updating usage data collection preferences 1054
Clearing out topology data without reinstalling 1055
Setting the idle session timeout 1058
Configuring the resources for topology for a production deployment 1060
Configuring probable cause 1061

IBM Cloud Pak for AIOps

Learn more

Overview

Getting started

Installation

Planning

System requirements

Installing

Preparing to install

Installing the Cloud Pak

Installing supporting products and extensions

Features

Data and tool connections

Application management

AI modelling and training

Topology viewing

User access control management

ChatOps insight management

Help and support

IBM Cloud Technology Slack channel

Troubleshooting

Support

IBM Cloud Pak® solutions closely align with other industry-leading software providers. While IBM values the use of inclusive language, terms that are outside of IBM's direct influence, for the sake of maintaining user understanding, are sometimes required. As other industry leaders join IBM in embracing the use of inclusive language, IBM will continue to update the documentation to reflect those changes.

Copyright IBM Corporation 2022. All Rights Reserved.

Release notes

Learn more about the IBM Cloud Pak® for AIOps documentation and what's new with IBM Cloud Pak for AIOps 4.2.0:

- What's new
- Known issues and limitations
- Regulatory compliance
- Deprecated and removed functions

What's new in IBM Cloud Pak for AIOps 4.2.0

Get a quick overview of what is new with IBM Cloud Pak® for AIOps.

IBM Cloud Pak for AIOps leverages AI to streamline IT operations, increase uptime, improve efficiency, and reduce costs. The solution supports Operations teams to deliver optimum performance and assurance by:

- Breaking down data silos
- Providing real-time insights into IT systems
- Augmenting incident management with AI
- Automating incident resolution

IBM Cloud Pak for AIOps 4.2.0 delivers new visualization tool interoperation to help improve Ops focus; user self-service and control of alerts, policies, and filters; and inventory filtering and search.

Important: Prior to Version 4.2.0, Cloud Pak for AIOps was known as Cloud Pak for Watson AIOps. With Version 4.2.0, the User Interface, IBM documentation, and other related documentation and materials are updated to use the name Cloud Pak for AIOps.

The following new features and changes are introduced with this IBM Cloud Pak offering:

- Install and upgrade
- Defining connections and integrations
- Automating processes (Policies, Runbooks, Actions)
- Metric anomaly detection
- Managing applications (services) and viewing topologies
- Resolving incidents and alerts
- Non-production (tech preview)
- Security and compliance

Install and upgrade

- Vault is no longer deployed from the AiopsEdge Operator.

Defining connections and integrations

- When you create an IBM Tivoli Netcool/Impact connection, the connector creates a Data Source name (DSA) in the IBM Tivoli Netcool/Impact Data Model (as before). In 4.2.0, the DSA name is created in the format of ***`AIOPs-ConnectorName-ConnectorID`***. Any IBM Tivoli Netcool/Impact policies will now use the new DSA instead of the old one. Therefore, you must update the new DSA with your cluster **Host Name** and **Authorization**. For more information, see Configuring the RESTful DSA data source.

Automating processes (Policies, Runbooks, Actions)

- You can now use complete URLs as parameters of **Client-side** actions and **HTTP** actions.
- The alert timeline now includes information about runbooks that have been invoked from within the context of the alert.
- Fully automated runbooks can now include predefined actions.

Metric Anomaly Detection

The Metric UI pages (Metric Search and the Anomaly UI) are updated:

- In Metric Search, the chart renders as zoomed in to 7 days of data, but you can zoom out to see all historical data. This applies to any KPIs added to the chart too.
- For the Incident Overview Panel there is now change, but when the Anomaly UI is launched from the panel, it renders as zoomed in to 7 days of data. You can zoom out to see all historical data. This applies to any KPIs added to the chart.

- For the side-panel in the alerts page there is no change, but when the Anomaly UI is launched from the panel, it renders as zoomed in to 7 days of data. You can zoom out to see all historical data. This applies to any KPIs added to the chart.
- You can display 1 baseline from all the KPIs that are in the related alerts and additional metrics tabs combined. This Additional metrics tab is only seen in the Anomaly UI.

Managing applications (services) and viewing topologies

- Deeper insight into data administration routines is available. You can click a routine name to open up a modal dialog that shows the parameter values that were used to run that routine the last time it was run. In addition, the main table footer now gives an indication of the number of routines being displayed, and in particular when some routines are not being shown due to your display settings, so you are aware that some routines are not shown.
- More filter conditions are available for searching and filtering resources (applications, resource groups, resources). You can now filter by the following conditions:
    - Severity: With this filter condition, you can filter based on checked items and their associated severity.
    - Business criticality: You can select a business criticality as a filter condition. Business criticality can be customized by an administrator.
    - Other properties: You can set more complex filter condition by clicking Add condition. See step 5 for more details.
    - For resources only, you can also filter by the following conditions to use date and time to filter resources:
        - Last updated: Use the last updated setting to view resources that were created or updated within the specified period prior to the current time.
        - Historical search: Use historical search to get data for a specific point in time that you set.

  For more information, see Resource management: Searching and filtering.
- You can now query the inventory service to find resources by status. For instance, a parameterized query for finding servers with a status of `critical` can resemble: `entityTypes=server AND alert_severity=critical`. For more information aboiut querying the inventory service, see Topology query syntax.

Resolving incidents and alerts

You can now see a timeline of how an alert or incident has progressed over time to help you discover what resolution activities were attempted for the alert or incident.

- For alerts, the timeline shows when an alert was created, its state, its severity changes, where it is assigned, whether it is acknowledged or suppressed, and any user comments.
- For incidents, the timeline shows when an incident was created, any stat chanages, where it is assigned, whether an alert is associated with it, and any user comments.

Non-production (tech preview)

IBM Cloud Pak for AIOps includes the following components and features for non-production use as a technology preview:

- Application disruption costs (in AIOps insights)
- Log anomaly automated training data selection
- Incident Creation based on Persistence and Golden Signal
- Feedback learning for statistical baseline log anomaly detection
- Log anomaly model drift detection
- Log anomaly detection golden signals
- Multi-zone high availability and disaster recovery (HADR)
- Datadog integration for event and topology data collection (observer and integration probe for data collection)
- Infrastructure Automation - Improved Automation with Embedded Workflows

For more information about these feature previews, see Non-production features (Technology preview).

Security and compliance

- **Fixed security vulnerabilities:** Fixes and updates are included to address reported security-related vulnerabilities with IBM Cloud Pak for AIOps. For more information about the included fixes, see Security bulletins and fixes.

Previously introduced features and changes

For more information about new features and changes that were introduced in previous versions of IBM Cloud Pak® for AIOps, see:

- What's new in Version 4.1.2
- What's new in Version 4.1.1
- What's new in Version 4.1.0
- What's new in Version 3.7.2
- What's new in Version 3.7.1
- What's new in Version 3.7.0
- What's new in Version 3.6.2
- What's new in Version 3.6.1
- What's new in Version 3.6.0
- What's new in Version 3.5.1
- What's new in Version 3.5.0

Known issues and limitations

Review the known issues for IBM Cloud Pak® for AIOps.

Also, review the troubleshooting documentation for more information on common issues. For more information, see troubleshooting.

- Install and upgrade
- Access control
- Observers and connections
- IBM Cloud Pak for AIOps and Netcool Operations Insights
- Applications and topologies
- Infrastructure automation
- UI console
- AI model management and training
- ChatOps
- Incidents and alerts
- Policies
- Secure Tunnel
- Runbook Automation
- AIOps Insights
- Ticketing

Install and upgrade

- ContainerStatusUnknown error
- Limitation on number of instances
- Manual resource adjustments are not persisted
- Services fail to connect to Cassandra
- The `ibm-aiops-orchestrator` pod throws an `OOMKilled` error
- Automatic approval required for installation
- Kong gateway pod is stuck in CrashLoopBackOff restarting issue
- Elasticsearch health status yellow after restoring from a backup
- ChatOps Microsoft Teams integration does not work with a proxy server
- Upgrade to 4.2.0 hangs, with IssueResolutionCore stuck in the "Update in progress" state
- The metric-spark pod restarts with an out of memory error
- Incorrect information appears on the Model versions tab of the Metric anomaly detection page after an upgrade

- Kafka topics cannot be listed with oc get kafkatopics
- Embedded EDB PostgreSQL database license key expiration

Also, review the troubleshooting documentation for more information on common issues. For more information, see troubleshooting.

ContainerStatusUnknown error

Spark pods might go into a **ContainerStatusUnknown** state with the **The node was low on resource: ephemeral-storage** error. The **ContainerStatusUnknown** state is due to Kafka being unhealthy. To work around this issue, ensure that Kafka is up and running by restarting the Spark pods.

1. Run the following commands to delete the Spark pods:

```
oc delete po spark-master
oc delete po spark-slave
```

Where **spark-master** is the name of the master pod and **spark-slave** is the name of the slave pod.

2. Confirm that the pods are restarted and are in a **Running** state, by running the following command:

```
oc get po | grep spark
```

Limitation on number of instances

IBM Cloud Pak for AIOps and Infrastructure Automation can co-exist on the same cluster, but you cannot have multiple instances of IBM Cloud Pak for AIOps or Infrastructure Automation on the same cluster.

Manual adjustments are not persisted

Custom patches, labels, and manual adjustments to IBM Cloud Pak for AIOps resources (such as increased CPU and memory values) are lost when an event such as upgrade, pod restart, resource editing, or node restart triggers a reconciliation. Reconciliation causes any manually implemented adjustments to be reverted to their original default values. Depending on the parameters that you want to adjust, you might be able to use a custom profile to persist your changes. For more information about custom profiles, see Custom profiles.

Services fail to connect to Cassandra

After you install IBM Cloud Pak for AIOps for a production environment deployment, various services might not be available due to connection issues with Cassandra. To resolve this issue if it occurs, restart Cassandra and the schema creation pods.

The `ibm-aiops-orchestrator` pod throws an `OOMKilled` error

If your environment has many secrets and ConfigMaps, when the **ibm-aiops-orchestrator** (lead operator) attempts to build its cache, the operator can exceed its memory allocation and cause a Kubernetes out-of-memory error for the container. This error can prevent the IBM Cloud Pak for AIOps installation from reconciling, blocking the installation from completing.

If you encounter this issue, the operator requires more memory resources to build its cache. Override the subscription resource to increase the memory limits for the pod and avoid the out-of-memory issue.

Automatic approval required for installation

The use of manual approval strategies for InstallPlans in a project (namespace) can affect the IBM Cloud Pak for AIOps installation.

For instance, if you use manual approval for any of your InstallPlans to install operators in **All Namespaces** mode (**cluster** scope), the manual approval can affect your install. The installation of IBM Cloud Pak for AIOps requires automatic approval to be used.

Kong gateway pod is stuck in CrashLoopBackOff restarting issue

In some case, the Kong gateway pod might have problems reaching a ready state. If this issue occurs, the Kong gateway pod can get stuck in **CrashLoopBackOff** and keep restarting. If you check the Kong gateway pod, you can see an error message similar to the following message:

```
bind() to unix:/usr/local/kong/stream_rpc.sock failed (98: Address already in use)
```

This issue occurs due to the Kong gateway proxy container nginx having a problem. To resolve this issue, manually delete the Kong gateway pod with the following command:

```
oc delete pod gateway-kong-xxxxxxxxx-xxxxx
```

Where **gateway-kong-xxxxxxxxx-xxxxx** is the name of the pod.

Elasticsearch health status yellow after restoring from a backup

When you are restoring an Elasticsearch backup to a new single node Elasticsearch cluster, the Elasticsearch database might not work as expected. Instead, the Elasticsearch cluster health shows that a yellow status after the restore completes.

ChatOps Microsoft Teams integration does not work with a proxy server

If you have an offline (air-gapped) deployment of IBM Cloud Pak for AIOps or an environment that uses a proxy server, then you cannot use the ChatOps Microsoft Teams connection. The use of a proxy with the Chatops Microsoft team connection is not supported.

Upgrade to 4.2.0 hangs, with IssueResolutionCore stuck in the "Update in progress" state.

When you upgrade to 4.2.0, an issue with the Redis operator can cause the upgrade to hang with IssueResolutionCore stuck in the "Update in progress" state. This issue does resolves without intervention. If you encounter this issue, wait for the issue to resolve, which can take up to a few hours.

The metric-spark pod restarts with an out of memory error

The **metric-spark** pod periodically restarts because it is out of memory. However, the system continues to run.

Incorrect information appears on the Model versions tab of the Metric anomaly detection page after an upgrade

After you upgrade IBM Cloud Pak for AIOps from version 4.1.2 to 4.2.0, the **Model versions** tab on the **Metric anomaly detection** page displays incorrect information for model versions that were created before the upgrade. The **Data starts** and **Data ends** columns display the same timestamp information as in the **Trained** column. The information displays correctly for the model versions that were created after the upgrade.

Kafka topics cannot be listed with oc get kafkatopics

Kafka topics are created by using the KafkaTopic custom resource or API. Only topics that are created by custom resource are shown when running **oc get kafkatopics**. To see a full list of all the Kafka topics use the following steps:

1. Install kcat.
2. Get the **waiops-mustgather.sh** script. For more information, see Installing the IBM Cloud Pak for AIOps MustGather tool.
3. Run **waiops-mustgather.sh -V kafka-topics**

Embedded EDB PostgreSQL database license key expiration

EDB PostgreSQL is a component of IBM Cloud Pak for AIOps. The EDB PostgreSQL database license key expires in October 2024, and will need to be refreshed before this time. For more information, see Embedded EDB PostgreSQL database license key expires on 1 October 2024.

Access control

- Automation Analyst role unused in IBM Cloud Pak for AIOps
- Service Administrators cannot manage others roles or view role details
- Users from a user group remain after user group is deleted
- UI redirects to unexpected page when logging in after a session timeout
- Users in a user group are not listed under the Manage assignees pane of the Incidents and alerts page

Automation Analyst role unused in IBM Cloud Pak for AIOps

By default, an Automation Analyst role is displayed within the IBM Cloud Pak Automation console Access control page when you are assigning a role to a user. This default role is used within the IBM® Automation family of offerings, which includes IBM

Cloud Pak for AIOps, however, this role is not used within IBM Cloud Pak for AIOps.

This role does not include or provide any permissions within IBM Cloud Pak for AIOps and should not be assigned to users within IBM Cloud Pak for AIOps.

Service Administrators cannot manage others roles or view role details

Users with the Service Administrator role do not have permission to add or update a role, or view the details of a users assign role. If a user with the Service Administrator role selects to view details about a role, a 401 error page is instead displayed.

Users from a user group remain after user group is deleted

When you delete a user group, the users that were included in the group remain in your list of users. Any role that is inherited through the deleted user group is removed from the users. If the users were assigned roles individually, they continue to have those roles and can continue to log in to the UI console and complete tasks. If the users that were in the deleted user group need to be removed completely, an administrator needs to manually remove the users. Users can be removed by clicking the *Delete* icon for the user's entry within the list of users on the *Access control Users* tab.

UI redirects to unexpected page when logging in after a session timeout

After a session timeout occurs and a user logs in to the UI console again, the user can be redirected to a different page than the page that they were on when the timeout occurred. For instance, a user that was working on the AI Model Management training page when their session timed out might be redirected to a graphql playground page after logging back in. This redirect occurs because the UI uses the last request URL that included the expired token to identify where to redirect the user when the user logs back in. If this redirect occurs, the user needs to manually return to the expected page in the UI to continue working.

Users in a user group are not listed under the Manage assignees pane of the Incidents and alerts page

When you have users within a user group and view the *Manage assignees* pane of the Incidents and alerts page, you might not see some users who are listed. This error can occur when the users from the LDAP user group are not individually onboarded. To verify whether a user is onboarded, go to the **Access control** > **Users** tab and check whether the user is listed. If the user is not listed, that user must first log in to the console, which validates their roles and permissions. After logging in, the user can display in the list of users and on the *Manage assignees* pane.

The *Manage assignees* pane is viewable from the list of all Incidents. Select an incident and then click *Manage assignees*. After you select an existing user group, you should see the included users who are listed.

Observers and connections

- ServiceNow Observer UI displays superfluous character
- Scheduled job for ServiceNow observer fails after upgrade
- 'Failed to read certificate' error
- Duplication of resources in the topology if certain observer job parameters are changed after a job has been run
- Incomplete historical data processing in the event of connector pods restarting
- File and Rest observer topology service location URL not accessible
- Connector console displays special characters incorrectly
- Turbonomic integration with IBM Cloud Pak for AIOps affects other integrations in Turbonomic
- No notification in IBM Cloud Pak for AIOps on Turbonomic actions closed without execution
- New Relic observer does not support dashboard tokens for new users
- All dates and time are in US-en format
- Appdynamic historical start date and time cannot be older than 4 hours
- Appdynamic live mode aggregation interval is 1 minute
- The observer-service pod is in crash back loop due to ghost vertex
- Alerts for Instana without associated topologies
- Alerts for Instana topology not mapping properly
- Instana metric collection API rate limit exceeded error
- IBM Cloud Pak for AIOps cannot close some alerts when an Instana connection exists
- Changing the codec for a connection can cause errors
- Delay query time for connections
- When creating a kubernetes observer job that is named 'weave_scope', 'load', 'kubeconfig', or 'local', the job fails to run

- Log Anomaly 8k limit on field mapping in details field of the alert schema
- Log connection does not start when multiple log connections are active
- After a restore, data from connections is not processed
- Turning on a disabled Dynatrace connection that collects live data results in an error
- Historical data from ServiceNow instance gets collected only when the historical data flow is reenabled
- Cannot edit or disable AppDynamic or Dynatrace connection due to 400 Bad Request error
- Dynatrace connector pod restarted and does not retrieve all historical data
- AppDynamics, Dynatrace, IBM Tivoli Netcool/Impact, Generic Webhook: unable to create a connector with special characters in name
- ServiceNow user account gets locked out after a few hours
- Scale resources when running log anomaly training on large data
- The connection status for Elk, Custom Logs, LogDNA, and Falcon LogScale sometimes show 'not running' even though the Flink job and gRPC pod are running correctly
- Log data connections status is "Done" even though historical data is still loading
- IBM Tivoli Netcool/Impact connection stops event processing with exceptions
- Impact connector fails for Impact server with non-default cluster name
- IBM AIOps Metric Manager processes historical data for only the first metric source
- Aggregation of metrics not working reliably for connectors, data loss occurring

ServiceNow Observer UI displays superfluous characters

If the Entity_mapping field is updated from the Observer UI, superfluous curly brackets and quotations are displayed in the Resource type-mapping field.

Workaround: This is a cosmetic issue and can be ignored.

Scheduled job for ServiceNow observer fails after upgrade

Following the latest upgrade, a previously scheduled ServiceNow Observer job enters an Error state when it runs.

Cause: An observer job was started with tables that are no longer recognized after the upgrade.

Workaround:

- On the *Data and tools connections* page, select the ServiceNow connection that was created before the upgrade and click **Edit** on the overflow menu.
- Navigate to the *Collect topology data (optional)* page and specify the values for the 'ServiceNow tables to be discovered by observer' and 'Maximum number of tables per cmdb_ci_rel api batch call' fields.
- Click **Next**, and then **Save** to save the changes. The job should now run correctly as scheduled.

'Failed to read certificate' error

This error can occur when an observer attempts to create an SSL certificate and the endpoint server does not respond.

In the example error message below, a vmvcenter.crt certificate error occurs because the endpoint server does not respond.

```
Failed to read certificate for field [certificate].
The file 'vmvcenter.crt' could not be found under /opt/ibm/netcool/asm/security/.
```

Workaround: Ensure that the endpoint server is running correctly.

Duplication of resources in the topology if certain observer job parameters are changed after a job has been run

Certain resource parameters are used to uniquely identify a resource. If one of these parameters is changed after the initial job run, then any subsequent job run will result in duplicate records. For example, if the parameter of 'hostname' is replaced with 'Ipaddress' after a topology has been created, a subsequent discovery will consider the resource as new, and create a duplicate record.

The following resource parameters uniquely identify a resource. Changing them after the initial job has been run will result in duplicate records.

Workaround: If you need to modify these values, do not modify the existing job. Instead, create a new job.

Table. Observer job parameters

Observer	Job parameter
ALM	n/a
AppDynamics	`account`
AWS	`region`, `dataTenant`
Ansible AWX	`host, user`
Azure	`data_center`
BigFix Inventory	`data_center`
Big Cloud Fabric	`proxy-hostname`, `proxy-username`, `bcf-controllers`
Ciena Blue Planet	`data_center`, `tenant`
Cisco ACI	`tenant_name`
DNS	`addressTypes`, `server`, `port`, `recurse`
Docker	`endPoint.port`
Dynatrace	`datatenant`, `hostname`
File	`provider`, `file`
GitLab	`datatenant`, `hostname`
GoogleCloud	`project_id`
HPNFVD	`datacenter`, `username`, `cnf_job`
IBM Cloud	`instance`, `username`, `region`
ITNM	`instance`
Jenkins	`jenkins_observation_namespace`
Juniper CSO	`cso_central_ms_url`, `user_domain_name`, `domain_project_tenant_name`
Juniper Contrail	`api_server_url`, `os_project_name`, `os_tenant_name`
Kubernetes	`data_center`, `namespace`
NewRelic	`accountName`, `accountId`
OpenStack	`data_center`, `os_project_name`
Rancher	`accessKey`, `clusterId`
REST	`provider`
SDC ONAP	`host`, `username`
ServiceNow	`instance_url`, `username`
SevOne	`datatenant`, `hostname`
TADDM	`api_url`, `username`
Viptela	`data_center`
VMware NSX	`data_center`
VMware vCenter	`data_center`
Zabbix	`data_center`

Incomplete historical data processing in the event of connector pods restarting

If you create a connection to collect historical data for Metric Anomaly AI Training, you might come across an issue where the connector pod restarts, but does not retrieve all historical data for training. As a result, you might suffer data loss.

A connector pod can restart due to outages, the target system crashing, or pod crashes in the environment. This issue can occur intermittently, depending on the number of metrics that are selected for the connection and the amount of data to be retrieved.

File and Rest observer topology service location URL not accessible

When creating an edge via either the File or Rest observers, the POST request returns a Topology service location URL that is not accessible. The URL cannot be used to manage the edge because the relevant API is not exposed. **Workaround:** None

Connector console displays special characters incorrectly

If you use special characters in the **Name** and **Description** fields while creating or editing a connection, the Connector console might display the special characters incorrectly. Nevertheless, the connection is saved.

Turbonomic integration affects other integrations in Turbonomic

The Turbonomic integration with IBM Cloud Pak for AIOps enables actions that are created or executed in Turbonomic to be notified in AIOps through the enabled webhook workflow. However, Turbonomic allows only one webhook workflow per action. Therefore, other integrations that are enabled in Turbonomic, like ServiceNow, might not get any notification when actions are created or executed in Turbonomic.

No notification in IBM Cloud Pak for AIOps on Turbonomic actions closed without execution

IBM Cloud Pak for AIOps does not receive any notification from Turbonomic for actions that are closed without being executed. For example, an action related to an erroneous condition that is no longer occurring gets automatically closed in Turbonomic. But its corresponding AIOps alert remains open indefinitely and must be cleared manually from the console.

New Relic observer does not support dashboard tokens for new users

For *new users* of the New Relic observer, the observer does not work as it no longer supports the New Relic One dashboard token. However, it will continue to work for *existing users* who are using the old token that was generated previously through the old dashboard.

All dates and time are in US-en format

When you are scheduling data collection for a connection, all dates and times are presented in the US-en formats:

- All dates are configured and presented in the `mm/dd/yyyy` format.
- All times are configured and presented in the `hh:mm AM/PM` 12-hour clock format.

You cannot switch the date or time format.

Appdynamic historical start date and time cannot be older then 4 hours

Historical start date and time is configurable, but if you set it to beyond the past 4 hours, then the connector will ignore it and only retrieve the past 4 hours of data.

Appdynamic live mode aggregation interval is 1 minute

In live mode, the new aggregation interval allowed is only 1 minute.

The observer-service pod is in a crash back loop due to a ghost vertex

If you notice that the topology observer-service pod is not functioning correctly and that restarting the pod does not correct the issue, a ghost vertex might need to be removed. To remove the vertex, you need to traverse an edge to the vertex, and then delete the vertex. To traverse to the vertex, use the type vertex and definesType edge.

1. Run the following command to find the ID for the type vertex.

   ```
   oc exec -it <topology pod> -- curl -X GET --header 'Accept: application/json' --header
   'X-TenantID: 01abea99-8dff-7f71-bef3-09136b6a4ff0'
   'https://localhost:8080/1.0/topology/types?
   _filter=keyIndexName=ASM::entityType::mgmtArtifact::ASM_OBSERVER_JOB' -u <username>
   <password> --insecure
   ```

   Where

   - `<username>` - Your Topology AIOps API username
   - `<password>` - Your Topology AIOps API password
2. Run the following command to use the definesType edge to get the ID for the vertex that is causing the issue.

   ```
   oc exec -it <topology pod> -- curl -X GET --header 'Accept: application/json' --header
   'X-TenantID: 01abea99-8dff-7f71-bef3-09136b6a4ff0'
   'https://localhost:8080/1.0/topology/resources/<type vertex
   ID>/references/out/definesType' <username><password> --insecure
   ```

   Where

   - `<type vertex ID>` - The ID for the type vertex that you retrieved in step 1.
   - `<username>` - Your Topology AIOps API username
   - `<password>` - Your Topology AIOps API password
3. Run the following command to delete the vertex.

```
oc exec -it <topology pod> -- curl -X DELETE --header 'Accept: application/json' --header
'X-TenantID: 01abea99-8dff-7f71-bef3-09136b6a4ff0'
'https://localhost:8080/1.0/topology/resources/<type vertex
ID>/references/out/definesType?_delete=nodes&_delete_self=false&_node_label=vertex' -u
<username><password> --insecure
```

Where

- `<type vertex ID>` - The ID for the type vertex that you retrieved in step 1.
- `<username>` - Your Topology AIOps API username
- `<password>` - Your Topology AIOps API password

Changing the codec for a connection can cause errors

When you are creating or editing a connection with the Data and tools connections tool, avoid changing the codec property value that is set in the mapping field. This property is not a configurable property. The IBM Cloud Pak Automation console sets the correct codec for the connection. If you change the value, the connection might fail, or it might not retrieve data for IBM Cloud Pak for AIOps. If you do need to set or change the value, ensure that you use the correct codec for the connection, such as `splunk` for a Splunk connection, `elk` for an ELK connection, or `Falcon LogScale` for a Falcon LogScale connection.

Search on the Add connection page does not function as expected

When you search on the Add connections page, you can view connectors that exist for other categories than the category under which you are searching. The search shows connectors that match the search query, regardless of the category of the connector.

Delay query time for connections

If you set a connection to retrieve *Live data for continuous AI training and anomaly detection* or *Live data for initial AI training*, you might need to configure a delay to offset the query time window to provide a time buffer for preventing the partial retrieval of real-time data. You cannot configure this delay within the UI console. You must use a command line to configure the delay. For more information about configuring this delay, see Delay configuration in data connections.

Alerts for Instana without associated topologies

In some cases Instana alerts will not have associated topologies. In most cases this happens because the resource that originated the event is no longer available in Instana. For example, a pod that triggers an Instana event can be redeployed by the underlying kubernetes engine.

Alerts for Instana topology not mapping properly

In some cases alerts are not mapped correctly to a corresponding Instana topology node. For example, alerts generated from log anomaly detection or metric anomaly detection (or other sources) might not show as associated with a Instana topology node.

As a workaround, you need to define your own match rules to correlate with the source data. To define a match rule, click on **Resource Management**, then click **Settings**, **Topology Configuration**, and finally **Configure** on the *Rules* Tile. When you are configuring the match token values to use, the values depend on the data that you are sending to Instana.

Instana metric collection API rate limit exceeded error

The recommended rate limit will be double the number of resources. There might be situations where the limits need to be increased. When a more precise limit is required, use the following formula to determine the limit to use:

```
number-of-metric-API-calls-per-hour ~= (number-of-selected-technologies x 2) x (snapshots-for-
selected-technologies / 30) x (60 / collection-interval)
number-of-topology-API-calls-per-hour ~= (number-of-application-perspectives x (60 /
collection-interval)) +(number-of-services x (60 / collection-interval))
number-of-events-API-calls-per-hour = 60

total= number-of-metric-API-calls-per-hour + number-of-topology-API-calls-per-hour + number-
of-events-API-calls-per-hour
```

Note: Each plugin can have a different number of metrics collected. The mean value across these is used, which is 2 collection cycles per plugin. If the environment is unbalanced, for instance you have mostly hosts that define most metrics, then the formula might underestimate the required limit.

To determine the number of resources (snapshots) for each infrastructure pluginAPI, use the following:

```
api/infrastructure-monitoring/snapshots?plugin=technology_name
```

Example:

```
api/infrastructure-monitoring/snapshots?plugin=host
```

For more information about Instana APIs, see Instana API.

The following example curl commands allow you to retrieve the number of:

- snapshots-for-selected-technologies (such as host)

  ```
  curl -k -s --request GET 'https://<instana server hostname>/api/infrastructure-monitoring/snapshots?plugin=host' --header 'Authorization: apiToken <api token>' | jq '.items|length'
  ```

- number-of-application-perspectives

  ```
  curl -k -s --request GET 'https://<instana server hostname>/api/application-monitoring/applications' --header 'Authorization: apiToken <api token>' | jq '.items|length'
  ```

- number-of-services

  ```
  curl -k -s --request GET 'https://<instana server hostname>/api/application-monitoring/services' --header 'Authorization: apiToken <api token>' | jq '.items|length'
  ```

IBM Cloud Pak for AIOps cannot close some alerts when an Instana connection exists

If you have an Instana connection created and have Instana 221 (Saas or Self-Hosted), you might encounter an issue where IBM Cloud Pak for AIOps might not be able to close some alerts. Instead, you might need to check the status for the associated event within the Instana dashboard and clear the alert manually. For more information, see Troubleshooting connections: Instana Event integration.

When creating a Kubernetes observer job that is named 'weave_scope', 'load', 'kubeconfig', or 'local', the job fails to run

If you create a Kubernetes observer job with the name **`weave_scope`**, **`load`**, **`kubeconfig`**, or **`local`**, the job always fails to run. When this error occurs, you can view an error icon in the schedule column for the job. To avoid this issue, do not use these names for the observer job.

Log Anomaly 8k limit on field mapping in details field of the alert schema

The limitation is that the datalayer imposes an 8 kb size limit on the details field in the alert schema. The details field is populated by the log anomaly event, which provides the relevant information to display in slack when trying to view alerts in the chatops. Whenever the details field size exceeds 8 kb, the returned json object is truncated and therefore when the user clicks **view alerts** to retrieve the alerts related to an incident, expected results are not seen and an error is recorded.

The current fields under the details objects are:

```
end_timestamp: int
original_group_id: str
causality: dict
detected_at: float
source_application_id: str
log_anomaly_confidence: float
log_anomaly_model: List[str]
prediction_error: dict
error_templates: List[int]
count_vector: List[int]
text_dict: dict
application_group_id: str
application_id: str
model_version: str
severity_from_model: int
description: str
```

Log connection does not start when multiple log connections are active

If you have many active log connections, such multiple Kafka, ELK, Splunk, and Falcon LogScale connections, and you create and enable another Falcon LogScale connection, you might notice that the connection status is stuck in an error or restarting state. This state can even occur after the connection is operating as expected.

This can occur if you exceed the limit for the number of jobs that can run on the underlying service, which results in insufficent resource available to start the connector. To resolve this issue, complete one or more of the following tasks:

- Increase the replica count of your task managers.
- Increase the task manager count per replica.
- Change the parallelism of your connections.
- Cancel other connections

After a restore, data from connections are not processed

If you have a connection that you are restoring, the status for these connections can be in error after the restore process completes. To resolve this status, you need to edit and save your connections with the Data and tool connections in the IBM Cloud Pak for AIOps IBM Cloud Pak Automation console. Editing the connection regenerates the associated Flink job for the connection, which updates the status.

Turning on a disabled Dynatrace connection that collects live data results in an error

If you have a connection to Dynatrace enabled for live data collection and then disable the connection, enabling the connection again can result in a **`java.lang.NullPointerException`** error occurring. If this error occurs, delete and then create the connection again to enable the Dynatrace data collection.

Historical data from ServiceNow instance gets collected only when the historical data flow is reenabled

If you enable historical data flow for a ServiceNow connection, you might notice that the historical data is not collected from ServiceNow. For instance, when you check the **`grpc-snow`** pod, you can see ticket data available, but when you check the Flink job or in Elasticsearch, you can notice that no data was collected. If this issue occurs, turning off the historical data flow and turning it back on can cause the data to begin to be collected.

Cannot edit or disable AppDynamic or Dynatrace connection due to 400 Bad Request error

If you have created an AppDynamic or Dynatrace connection, you might not be able to disable the connection or edit the collection mode, such as to change the mode from historical to live or live to historical. If this issue occurs, a **`400 Bad Request error`** message displays when you attempt to disable or edit the connection. Instead of disabling the connection, delete the connection and create a replacement connection when needed. As a workaround if you cannot edit the connection, you can create a replacement connection with your preferred settings.

Dynatrace connector pod restarted and does not retrieve all historical data

If you have a Dynatrace connection created and pull historical data with multiple metrics for Metric Anomaly AI Training, you can encounter an issue where the Dynatrace pod restarts, but does not complete retrieving the expected historical data for training. This issue can occur intermittently, depending on the number of metrics that are selected for the connection and the amount of data to be retrieved.

If this potential out-of-memory or out-of-resources issue occurs, consider creating separate connections to monitor different and smaller sets of metrics. By splitting the connections, you can reduce the amount of data to be retrieved through the initial connection that can cause this issue.

AppDynamics, Dynatrace: unable to create a connector with special characters in name

You cannot use spaces or special characters for names for AppDynamics and Dynatrace connectors. You can only use alphanumeric values.

ServiceNow user account gets locked out after a few hours

If there is an active ServiceNow connection with data collection enabled and the ServiceNow credentials change, the ServiceNow user account can get locked out. ServiceNow has an automatic login locking script called "SNC User Lockout Check", which locks users out after more than 5 failed attempts (including any failed API calls).

If you check the Incidents and alert page, you will see also an alert saying "ServiceNow instance authentication failed".

When this problem occurs, unlock the user in ServiceNow. Then change the password in the ServiceNow connection and save. When authentication fails in the ServiceNow connector, there is a 1-minute wait time before you can access it, to prevent a lockout from occuring quickly.

Scale resources when running log anomaly training on large data

In some cases it is observed that log anomaly training fails on large data due to being Out Of Memory (OOM) or if there is a problem with ES shards. The solution is to scale up the resources to handle large data training.

For more information about shard management, see About indices and shards. For more information about increasing ES Resources, see Log anomaly training pods CPU and Memory resource management.

The connection status for Elk, Custom Logs, LogDNA, and Falcon LogScale sometimes shows 'not running' even though the Flink job and gRPC pod are running correctly

On creating a connection, the Flink job retrieves data normally and the gRPC pod is running without error. However, the console shows that the connection status is 'not running'.

Log data connections status is "Done" even though historical data is still loading

When a log data connection (Falcon Logscale, ELK, LogDNA, Custom, Splunk) is running in **Historical data for initial AI training** mode, and a custom regex is added in the **`field_mapping`** section, the data processing can take a long time. Although the **Data collection status** might be shown on the UI as **Done**, data could still be being processed and written to Elastic in the background.

To speed up this process, you can increase the **Base parallelism** number that is associated with that connection. For more information, see Increasing data streaming capacity.

IBM Tivoli Netcool/Impact connection stops event processing with exceptions

If you have an IBM Tivoli Netcool/Impact connection, you can encounter an issue where the connection temporarily stops processing during the sending of an event stream to IBM Cloud Pak for AIOps.

This issue can occur when you have an IBM Cloud Pak for AIOps policy that triggers an IBM Tivoli Netcool/Impact policy when certain types of events are received. If this issue occurs and stops the event processing, the Impact connector logs or Impact policylogger logs can include messages that are similar to the following example exceptions:

```
[6/14/23, 11:38:45:816 UTC] 0000005d ConnectorMana W failed to send status update
...
[6/14/23, 11:38:45:815 UTC] 000023ca StandardConne W configuration stream terminated with an
error
...
[6/14/23, 11:38:45:816 UTC] 000023cc GRPCCloudEven W consume stream terminated with an error:
channel=cp4waiops-cartridge.lifecycle.output.connector-requests
```

If you encounter this issue, you might need to restart the **`impact-connector`** pod to begin the processing of the event stream again.

Impact connector fails for Impact server with non-default cluster name

If the Netcool/Impact cluster uses a non-default cluster name ("NCICLUSTER"), the connector may fail to validate the connection. The Impact server may report **`DynamicBindingException`** errors in the impactgui.log:

```
com.micromuse.common.nameserver.DynamicBindingException: DynamicBindingException: Service
[NCICLUSTER] not in nameserver.
```

To resolve the issue, wait for the backend Impact server to finish initializing before starting or restarting the Impact GUI server.

If Netcool/Impact is running fix pack 7.1.0.26 or later, you can also resolve the issue by setting the **`nameserver.defaultcluster`** property in the GUI server. Add the following line to *$IMPACT_HOME/etc/nameserver.props*:

```
impact.nameserver.defaultcluster=CLUSTERNAME
```

where *CLUSTERNAME* is the name of the Impact cluster.

IBM AIOps Metric Manager processes historical data for only the first metric source

Metric Manager processes historical data for only the first metric source. For example, if you turn on a metric source, such as log comprehension, the system starts to send metric data. Then, if you configure another metric source to load historical data, the historical data doesn't load.

Aggregation of metrics not working reliably for connectors, data loss occurring

There is currently a known issue whereby gaps may occur in metric data after a few days of metric data collection when running in live mode. Historical data collection does not appear to be impacted. This is noticeable in the Metrics Search UI, but the data itself (even that for the gaps) will be in Kafka.

IBM Cloud Pak for AIOps and Netcool Operations Insights

- IBM Cloud Pak for AIOps Strimzi Kafka topics created without replication
- IBM Cloud Pak for AIOps pods not starting after a cluster restart
- Error when accessing the AI Model Management
- NOIHybrid incorrectly listed in provided APIs for IBM Netcool Operations Insight operator
- Rare issue: unable to deploy log anomaly detection model

IBM Cloud Pak for AIOps Strimzi Kafka topics created without replication

IBM Cloud Pak for AIOps supports multiple replications of Kafka topics for large production installations, such as for data redundancy. The IBM Cloud Pak Automation console can automatically create Kafka topics when connections are created. When a topic is dynamically created by the IBM Cloud Pak Automation console, the replication is set to **1** in the controller. As such, Kafka topics the are created during installation can have multiple replicates, but those topics that are created dynamically do not.

If you are implementing a production (large) deployment of IBM Cloud Pak for AIOps, you might lose data if your Kafka pods fail or restart. If the data flow is enabled in your Kafka connection when the Kafka pods go down, you might experience a gap in the data that your connection generated during that down period. Upgrades or updates to workers can cause a Kafka broker restart.

You can manually modify the Kafka topic replication inside the Kafka container from a value of **1** to **3** to mitigate any potential data loss from this issue.

IBM Cloud Pak for AIOps pods not starting after a cluster restart

When the cluster is restarted, all nodes have a *STATUS* of *Ready*. Most pods return to a *STATUS* of *Running* except for some IBM Cloud Pak for AIOps pods.

One potential cause is that Elasticsearch must be up and running before the IBM Cloud Pak for AIOps pods can start.

Restart the Elasticsearch pod to get all pods back to a *STATUS* of *Running*.

Error when accessing the AI Model Management

The AI Model Management can fail to load when you click to open the tool from the *Quick navigation* links on the *Home* page.

This can occur if a network disruption occurred during installation. This disruption can result in the cluster becoming inaccessible and cause some steps to be missed during the platform UI startup. This disruption can result in expected Nginx rules to be missing.

To check whether the rules are missing, complete the following steps:

1. Open a command line and connect to your cluster with the **`oc login`** command.
2. Use the **`oc project`** command to set the context to the project where IBM Cloud Pak for AIOps is deployed.
3. Locate an **`ibm-nginx`** pod within your deployment:

   ```
   ibm_nginx_pod=$(oc get pods | grep ibm-nginx | head -1 | cut -f1 -d\ )
   echo $ibm_nginx_pod
   ```

4. Run the following **`oc exec`** command for one of the **`ibm-nginx`** pods to run commands against that pod.

```
oc exec $ibm_nginx_pod -- nginx -T | grep aimodel
```

After the command runs, check whether there is an ingress rule or entry for **aimodels**. Your output can resemble the following sample output:

```
                        location ~* (/aiops/([^/]+)/aimodels|/aiops/aimodels) {
nginx: the configuration file /usr/local/openresty/nginx/conf/nginx.conf syntax is ok
nginx: configuration file /usr/local/openresty/nginx/conf/nginx.conf test is successful
```

If you do not view an **aimodels** rule, complete the following steps to add the rule.

5. Open the platform UI extension configuration for the AI Model Management for editing:

   ```
   oc edit cm aiops-ai-model-ui-zen-extension
   ```

6. Increment the **icpdata_addon_version** version metadata label within the configmap to be **3.3.2**.

   ```
   icpdata_addon_version: 3.3.2
   ```

NOIHybrid incorrectly listed in provided APIs for IBM Netcool Operations Insight operator

NOIHybrid is incorrectly included in the **Provided APIs** list for the *Netcool Operations Insight* operator. This list is displayed in the Red Hat OpenShift Container Platform web console under **Installed Operators** > **Netcool Operations Insight** > **Operator Details**. Do not use *NOIHybrid* APIs.

Rare issue: unable to deploy log anomaly detection model

Very occasionally, following successful completion of log anomaly detection training, an error similar to the following is displayed in the AI management UI training page following an attempt to deploy the model.

```
Error
Model deployment failed
```

Within the error textbox, you will also see the text "**Forbidden**".

If you investigate the **aiops-ai-model-ui** pod logs, you will also see the following error.

```
ForbiddenError: invalid csrf token
```

If this occurs, first refresh the browser and try to deploy again.

If that does not remedy the situation, then log out and log back in, and then try to deploy the model again.

Applications and topologies

- Resources with a _compositeId value have 'Related Services' and 'Related resource groups' tabs disabled
- JVM heap out-of-memory (OOM) failures when loading large number of resources
- Critical error message displayed when attempting to render an application
- Different date and time in Automation console and ChatOps between users
- Deleting a tag template can cause out-of-memory errors
- The Find Path tool ignores filters
- Fault localization and blast radius are not producing accurate results
- High volumes of data can cause Spark workers to run out of memory
- Probable cause is not producing accurate results
- Azure observer missing subnet relationship in topology
- Topologies not visible on Incident Topology page after resource merge
- Openstack observer missing edge-runsOn connectivity in topology

Resources with a _compositeId value have 'Related Services' and 'Related resource groups' tabs disabled]

Group and service information cannot be fetched for resources that are part of a composite.

This defect is only encountered when viewing the related service or related resource group details of a resource that is part of a composite. This resource will have the 'Related services' and 'Related resource groups' tabs disabled.

Workaround: none

JVM heap out-of-memory (OOM) failures when loading large number of resources

When running topology loads in quick succession, it is possible to experience some OOM errors and undesired topology pod restarts, even though the pods will continue the processing after restarting.

This error can occur when running resource loads of several millions in a large deployment, and could slow down the loading process. The following type of error message can be seen in the pod logs:

```
WARN [2022-10-25 15:43:31,906] [JanusGraph Session-io-4] c.d.o.d.i.c.c.CqlRequestHandler -
Query '[4 values] SELECT column1,value FROM janusgraph.graphindex WHERE key=:key AND
column1>=:slicestart AND column1<:sliceend LIMIT :maxrows
[key=0x02168910cfd95b7e3bc74006a4a8a73a79c71255a0726573...<truncated>, slicestart=0x00,
sliceend=0xff, maxrows=2147483647]' generated server side warning(s): Read 5000 live rows and
1711 tombstone cells for query SELECT value FROM janusgraph.graphindex WHERE key =
02168910cfd95b7e3bc74006a4a8a73a79c71255a07265736f757263e5 AND column1 >
00392470118001287159029001287150029 0 AND column1 < ff LIMIT 5000; token 9157578746393928897
(see tombstone_warn_threshold) JVMDUMP039I Processing dump event "systhrow", detail
"java/lang/OutOfMemoryError" at 2022/10/25 15:43:32 - please wait. JVMDUMP032I JVM requested
System dump using '/tmp/cassandra-certs/core.20221025.154332.1.0001.dmp' in response to an
event
```

Cause: Not enough headroom exists between JVM memory limit and the pod memory limit, usually because one was increased without also increasing the other.

Workaround: Ensure that any changes in heap size maintain enough headroom between these settings.

Example: In this example (for a topology size1) the pod limits are set to 3.6 GB while the maximum memory for the JVM (**-Xmx**) is set to 3 GB, thereby leaving 0.6 GB of headroom free for use by the OS.

```
size1:
   enableHPA: false
   replicas: 2
   jvmArgs: "-Dcom.ibm.jsse2.overrideDefaultTLS=true -Xms1G -Xmx3G"
   resources:
      requests:
         memory: "1200Mi"
         cpu: "2.0"
      limits:
         memory: "3600Mi"
         cpu: "3.0"
```

Critical error message displayed when attempting to render an application

This problem occurs when all of the groups, within the application you are attempting to render, have no members. When the application is selected in **Application management**, it does not render and a critical error message is displayed on the UI.

Avoid creating applications with no members. If an application with no members was created for test purposes only, then ignore this error.

Different date and time in Automation console and ChatOps between users

The date and time format for an Incident in the IBM Cloud Pak Automation console *Application management* tool and the associated ChatOps notification can be different between users. The format and time zone that is used in the Automation console and ChatOps notification is set to the user's locale. If different users are in different time zones, the displayed date and time are different in the Automation console and ChatOps notification.

Deleting a tag template can cause out-of-memory errors

If a tag is applied to a large number (that is, thousands) of topology resources, then deleting the tag template can cause out-of-memory errors with the **topology-merge** pod.

Avoid creating tag templates that use tags that occur with such frequency. Later, do not tag thousands of resources with the same tag, and avoid them being used in a group.

The Find Path tool ignores filters

The topology path tool fails to launch with filters applied.

Launch the path tool without filters, then manually apply the filter settings on the path page.

Probable cause is not producing accurate results

The correlation algorithms for probable cause currently require the use of a Kubernetes model with service-to-service relationships, or the use of dependency relationships between non-Kubernetes resources.

Complete the following steps to create the required relationships for Kubernetes. This procedure configures Topology Manager to overlay relationships provided by the File observer onto the Kubernetes topology.

Note: The Kubernetes observer must be configured and loading data.

1. Log in to the IBM Cloud Pak Automation console.
2. From the main navigation, expand **Operate** and click **Topology viewer**.
3. From the topology navigation toolbar, expand **Settings**, click **Topology configuration**.
4. On the **Rules** tile, click **Configure** to navigate to the *Rules administration* page.
5. On the **Merge** tab, click **New** to create a **New merge rule**.

   In this scenario, data that is provided by the File observer will be used to add the relationships.
6. Specify the following information on the **New merge rule** page:
   1. Rule name: **`k8-file-service`**.
   2. Set the rule Status to **Enabled**.
   3. Add the **`uniqueId`** property to the set of Tokens.
   4. Expand the **Conditions** section and select **File** and **Kubernetes** from the set of available Observers and click **Add**.
   5. Specify **`service`** for **Resource types** and click **Add**.
   6. Click **Save** to save the new Merge rule.
7. Locate the services that you want to relate together and make a note of their source-specific uniqueId, such as **`05f337a1-5783-43bb-9323-dfba941455c7`** (shipping) and **`ae076382-3df9-46cb-97e9-a0342d219efb`** (web).
8. Create a file for the File Observer that contains the **`service-dependsOn-service`** relationships necessary for the correlation algorithms to work.

   The following example creates two services, web and shipping, and states that web **`dependsOn`** shipping. Repeat this as required to relate your services together.

   ```
   V:{"uniqueId": "05f337a1-5783-43bb-9323-dfba941455c7", "name": "shipping", "entityTypes":
   ["service"]}
   V:{"uniqueId": "ae076382-3df9-46cb-97e9-a0342d219efb", "name": "web", "entityTypes":
   ["service"]}
   E:{"_fromUniqueId":"ae076382-3df9-46cb-97e9-a0342d219efb", "_edgeType":"dependsOn",
   "_toUniqueId":"05f337a1-5783-43bb-9323-dfba941455c7”}
   ```
9. Load this file into Topology Manager to relate the services. For more information, see Configuring File Observer jobs.

   If your topology changes, then re-create and reload the file as required. A similar process can be followed for non-Kubernetes sources.

High volumes of data can cause Spark workers to run out of memory

If your environment handles high (10+ millions) workloads of alerts or event, your Spark workers can run out of storage (ephemeral storage). If you encounter this issue, restart the affected Spark workers. This issue can also occur if you are running multiple jobs, which can cause the file system to fill, such as with log or JAR files.

Azure observer missing subnet relationship in topology

For the Azure Observer, a subnet can be intermittently missing the relationship with an IP address in the topology for a resource. While the relationship can be intermittently missing, both the subnet and IP address verticies remain available in the topology.

Topologies not visible on Incident Topology page after resource merge

When resources from two observer sources have been merged using the topology merge functionality, the topology is no longer displayed in the Incident view. This known issue affects **only** the Incident view, and the topology is still present in all other views.

Openstack observer missing edge-runsOn connectivity in topology

After running an Openstack observer job, the **`edge-runsOn`** connectivity between **`ComputeHost`** and **`Hypervisor`** elements is not shown in **`Resource Managenent -> Resources`** when it should be.

Infrastructure Automation

- Kubernetes permissions are missing for user roles for using Managed services and the Service catalog
- Non-LDAP users cannot access Infrastructure management
- Red Hat Advanced Cluster Management and IBM Cloud Pak for Multicloud Management core are not supported
- Users are redirected to the Administration panel when logging back into the UI
- Managed services secrets and certificates cannot be customized during installation
- Database fails to reset when error occurs during database creation for Infrastructure management
- The **`cam-tenant-api`** pod is not in a ready state after installing the iaconfig CR
- Print or export as PDF entire tables does not work as expected
- Infrastructure management log display in the UI is removed
- Infrastructure Automation Test deploy fails
- On Infrastructure management appliances, an Ansible playbook deployment fails
- After restoring from a backup, the Managed services deployment fails
- Infrastructure management fails to save container provider after changing to a new token

Kubernetes permissions are missing for user roles for using Managed services and the Service catalog

If you install Infrastructure Automation, you, or an administrator, must add the required Kubernetes permissions to user roles before your users can begin to access and use Managed services or the Service catalog.

As an administrator, add the following permissions to your use roles:

Table. Required permissions

Role	Required permission for Infrastructure Automation
Automation Administrator	Administer Kubernetes resources
Automation Operator	Manage Kubernetes resources
Automation Developer	Edit Kubernetes resources
Automation Analyst	View kubernetes resources

For more information about how to add permissions to a role, see Managing roles for Infrastructure Automation.

Non-LDAP users cannot access Infrastructure management

Non-LDAP authenticated users cannot be used with Single Sign-On for Infrastructure management. If you attempt to use Infrastructure management with a non-LDAP authenticated user, you can encounter the following error:

While logged in to the Infrastructure Automation UI console with a non-LDAP user, attempting to start Infrastructure management fails with an error. This is a limitation.

The error states:

```
OpenID Connect Provider error: Error in handling response type.
```

Red Hat Advanced Cluster Management and IBM Cloud Pak for Multicloud Management core are not supported

Installation of Infrastructure management in IBM Cloud Pak for AIOps does not support Red Hat Advanced Cluster Management and IBM Cloud Pak for Multicloud Management core. You can continue to use the Kubernetes cluster life-cycle templates and services to create a Kubernetes cluster and import the cluster to an existing installation of Red Hat Advanced Cluster Management, if an installation is available. Deploying hybrid applications are also not supported by Infrastructure Automation.

Users are redirected to the Administration panel when logging back into the UI

When you are working within Infrastructure Automation and log out and then log back in, you can be redirected to the Administration panel instead of the Infrastructure Automation home page or other page that you were previously using. If this occurs, you can use the Cloud Pak switcher in the upper right of the UI console to switch to the Infrastructure Automation home page and then return to the page that you were previously using.

Database fails to reset when error occurs during database creation for Infrastructure management

If you are creating the database for the Infrastructure management appliance and you encounter an error, such as the database creation failing to complete successfully, you might not be able to continue with your setup without redeploying. For instance, if the creation fails, resetting the database to clean up your database and deployment can also fail. To resolve this issue, you need to redeploy the Infrastructure management appliance image before reattempting to create the database.

The `cam-tenant-api` pod is not in a ready state after installing the iaconfig CR

After you install Infrastructure Automation, you can encounter an error where the **`cam-tenant-api`** pod displays as running, but not in a ready state. When this error occurs, you can see the following message:

```
[ERROR] init-platform-security - >>>>>>>>>> Failed to configure Platform Security. Will retry
in 60 seconds <<<<<<<<<<< OperationalError: [object Object]
```

If this error occurs, delete the **`cam-tenant-api`** pod to cause the pod to restart and attempt to enter a ready state.

Print or export as PDF entire tables does not work as expected

If you are using the Firefox browser, and you select **Print or export as PDF** on the **Compute** > **Containers** > **Projects** page to print or export the entire table of data, the print, or export might not work as expected. For instance, some data, such as table rows might be missing. If you encounter this issue, try a different browser for printing or exporting the data.

Infrastructure management log display in the UI is removed

Log display support on the UI is removed for Infrastructure management. As an alternative for viewing these logs, use Kubernetes standard methods such as **`oc log`** commands, viewing the output in Red Hat OpenShift Container Platform or Kubernetes, or setting up a log aggregator for your cluster.

You can still see the log tabs (**Collect Logs**, **IA:IM Log**, **Audit Log**, and **Production Log**) on the **Settings** > **Application Settings** **Diagnostic** page. However, instead of displaying the log information, the following message is displayed: **`Logs for this IA:IM Server are not available for viewing`**.

Infrastructure Automation Test deploy fails

Infrastructure Automation Test deploy from a **Service Overview** page fails to deploy.

On Infrastructure management appliances, an Ansible playbook deployment fails

When you attempt to deploy an Ansible playbook on an Infrastructure management appliance through an embedded Ansible deployment, the playbook deployment can fail with the following error:

```
<35.237.119.31> ESTABLISH SSH CONNECTION FOR USER: ubuntu
fatal: [35.237.119.31]: FAILED! => {
"msg": "Unable to create local directories(/home/manageiq/.ansible/cp): [Errno 13] Permission
denied: b'/home/manageiq'"
}
```

If you encounter this error, log in to the appliance as the root user and then deploy the playbook again:

1. Run the command:

   ```
   mkdir -p /home/manageiq
   ```

2. Run the command:

   ```
   chown manageiq:manageiq /home/manageiq
   ```

3. Deploy the Ansible playbook again.

After restoring Managed services from a backup, the Managed services deployment fails

After you restore Managed services (**cam**) from a backup, the deployment instance fails with a **socket hang up** error.

If this error occurs, restart the **cam-iaas** pod by running the following command:

```
oc delete pod <cam-iaas-xxxx> -n <namespace>
```

Where **<namespace>** is the project (namespace) where Infrastructure Automation is installed, and **<cam-iaas-xxxx>** is the name of the **cam-iaas** pod to restart.

With this restart, the service deployment can complete successfully.

Infrastructure management fails to save container provider after changing to a new token

Infrastructure management fails to save changes to a container provider after updating the access token when the Metrics collection is enabled.

Workaround: After updating and validating the new token in the *Edit provider* dialog box, switch to the **Metrics** tab and validate the existing endpoint. The **Save** button is now enabled.

UI console

- Tour icon disappears when browsing to another console page while a guided tour is running
- About page does not show the correct version of IBM Automation
- 'Data cannot be displayed': error given for 'Defined Applications' and 'Favorite Applications' tiles on the Home page
- User credential timeout starts in the backend

Tour icon disappears when browsing to another console page while a guided tour is running

When you start a guided tour and navigate away from the page to the IBM Automation *Home* page, the *Tour* icon might not display on the toolbar. This behavior can occur when an IBM Cloud Pak for AIOps tour is still running. Only one guided tour can run at a time. To resolve this issue, return to an IBM Cloud Pak for AIOps page, click on the *Tour* icon and close a tour. When you return to the IBM Automation *Home* page, the *Tour* icon reappears.

About page does not show the correct version of IBM Automation

If you are attempting to identify the version of an installed IBM Cloud Pak for AIOps, the *About* page that is accessed from the console toolbar does not show the correct version.

To determine the correct version, use the Cloud pak switcher to access the **IBM Cloud Pak | Administration** tool. On this tool, find the **Cloud Pak deployment** summary card and click **View details**. The side panel opens. Expand the entry for **IBM Automation** to view the installed instances. The details for the installed instance shows the current deployed version of the IBM Cloud Pak.

'Data cannot be displayed': error given for 'Defined Applications' and 'Favorite Applications' tiles on the Home page

On opening the Console home page neither 'Defined Applications' nor 'Favourite Applications' is listed in their tiles. However, they do exist as they can be viewed under the 'Resource Management' section. The fix for this is to restart the aiops-base-ui pods.

User credential timeout starts in the backend

The first indications of a user credential timeout might be backend failures. For example, failure to load incident, alert, or policy lists. To resolve this issue if it occurs, log out and log back in again, or wait for the frontend logout to occur.

AI Model management and training

- Log parsing assigns messages to catch-all template instead of generating expected template
- Elasticsearch record count does not match record count published to Kafka topic
- Log anomaly detection EXPIRY_SECONDS environment variable not retained after an upgrade
- Log anomalies are not detected by natural language log anomaly detection algorithm
- Metric anomaly detection training does not run on schedule
- In Change risk training, Precheck indicates “Good data” but models fail to create

Log parsing assigns messages to catch-all template instead of generating expected template

If you use catch-all templates for mapping uncategorized messages during AI model training, you can encounter an issue where the log parsing assigns messages for an error to the catch-all templates instead of generating an expected template for that error. If this issue occurs, you might not see expected anomalies.

If you suspect this issue is occurring and you do not see expected anomalies, complete the following steps to manually verify your training templates, and remove any catch-all templates that incorrectly generated.

1. Retrieve the normalized logs from your logtrain indices.
2. Identify the logs that are error logs. Review those logs to determine the template mappings.
3. Retrieve the identified templates from Elasticsearch.
4. Use the error log contents and the template ID from the retrieved normalized logs to identify the template string within the retrieved templates.
5. If the template string is comprised entirely of parameters, or a single word and parameters, the template might be a catch-all template. For example, the following string is an example of a catch-all template:

   ```
   <>to <><><><><>
   <> <> <> <>-<>-<> <> <> <> <> <> <> <> <> <> <>
   ```

6. Manually delete any catch-all templates.

Elasticsearch record count does not match record count published to Kafka topic

When you push a large training file (for example 60 M records, such as logs or events) to Kafka through your configured connection, the number of records that are ingested and displayed on Elasticsearch might not match. Elasticsearch record count might be lower than Kafka count due to deduplication. If you encounter this issue, split large files into smaller batches and send them individually to Kafka (for example, 5 M records each). When you are pushing a batch, ensure that you wait for an ingest to complete and the associated records display on Elasticsearch before you push the next batch of records.

Log anomaly detection EXPIRY_SECONDS environment variable not retained after an upgrade

If you set a value for the *EXPIRY_SECONDS* environment variable and upgrade, the environment variable is not retained after the upgrade.

After the upgrade is completed, set the environment variable again. For more information about setting the variable, see Configuring expiry time for log anomaly detection alerts.

Log anomalies are not detected by natural language log anomaly detection algorithm

In some cases a model that has been trained successfully is unable to detect certain log anomalies. The quality of the model is independent of whether it trained successfully, and model quality tends to improve as more training data is available. If the model is not detecting anomalies in your logs, consider training the model again but using additional days of training data to improve the model quality.

Metric anomaly detection training does not run on schedule

If you have metric anomaly detection training scheduled to run, such as daily, you can encounter an issue where the training does not run as scheduled. If the training job does not run on schedule, log in to the IBM Cloud Pak Automation console and

click the *Metric anomaly detection* algorithm tile and then *Train models*.

In Change risk training, Precheck indicates “Good data” but models fail to create

On rare occasions, a Change risk model fails to create, even though *Precheck data* indicates that the data is good. This failure is caused by an insufficient number of problematic change risk tickets being available to create a good model. This problem resolves itself when enough tickets become available for the model. (For more information, see Closed change ticket count requirements).

To confirm that insufficient problem tickets is causing the failure, view the Change risk logs on the Luigi pods.

To retrieve the pod:

```
oc get pod | grep luigi-cr
```

View the logs for training Change risk models.

```
oc logs <pod-name> # Ex: luigi-cr-1b5ef57f-9053-4037-95ca-c1e8b8748fc5
```

Check whether the log contains the following

```
size of the problematic (aka labels) tickets is insufficient
```

If confirmed, ensure that enough problematic change tickets are available before training the model again.

ChatOps

- In Slack, a ChatOps communication to IBM Cloud Pak for AIOps times out without establishing a connection
- In a Microsoft Teams ChatOps, the attach template logs feature does not work
- Incidents cannot be reopened or restored in ChatOps
- No Recommended runbooks found in incident overview
- No incident content viewable in Microsoft Teams on mobile devices
- During a ChatOps secure tunnel creation an 'installation failed' message displays

In Slack, a ChatOps communication to IBM Cloud Pak for AIOps times out without establishing a connection

When sending a ChatOps communication from Slack to IBM Cloud Pak for AIOps, a known intermittent issue can occur. The communication between Slack and IBM Cloud Pak for AIOps can time out after 3 seconds of no response. A potential solution is to reconfigure your connection in the connection onboarding. Ensure that your Slack Bot can access your IBM Cloud Pak for AIOps instance in a prompt fashion. A more permanent and robust solution to this issue is being devised.

In a Microsoft Teams ChatOps, the attach template logs feature does not work

If you have a Microsoft Teams ChatOps, clicking the **Attach template logs** button does not work and the logs are not sent to your Microsoft Teams channel for review. As an alternative, use the **Preview logs** button to view the template logs.

Incidents cannot be reopened or restored in ChatOps

When an incident is closed it is archived and can no longer be modified. If a new alert occurs that is related to the archived incident, a new incident is created instead of reopening the archived incident.

No Recommended runbooks found in incident overview

If a runbook recommends to remediate an incident is deleted from the runbook Library, it does not remove the **Recommended runbooks** link in a ChatOps notification. This can result in the ChatOps runbook section linking out to an empty runbooks page in the incident overview.

No incident content viewable in Microsoft Teams on mobile devices

On mobile devices, when viewed in Microsoft Teams, incidents can appear with no viewable data. Where this happens, switch to using a computer to see the full incident data.

During a ChatOps secure tunnel creation an 'installation failed' message displays

When you create a ChatOps connection and it fails, wait for a few minutes to see if the installation retries, and if it does not simply create the connection again.

Incidents and alerts

- "An error occurred while fetching data from the server"
- Alerts tab shows "An unknown error occurred" error when all alerts are closed
- In the metric anomaly details chart PNG and/or JPG export does not appear to work
- Closed incidents are missing details and displaying a critical error
- Metric anomaly chart unexpectedly changes from zoom view to normal view
- Unable to add metric anomaly in Related alerts to chart
- Limitation of preview text for default recommended action
- Alert right-click menu items cannot be added if any actions have parameters with type 'array'
- Metric search page chart lines are disjointed

"An error occurred while fetching data from the server"

When viewing the Incidents UI or when creating a policy to assign runbooks to alerts, you might see the message "An error occurred while fetching data from the server". Or in the Alert Viewer, you might see "An unknown error occurred". If you encounter these error messages, complete the following steps to delete the **`aiops-ir-ui-api-graphql`** pod. The pod is then automatically re-created, which should resolve the error.

1. Log in to your cluster by running the oc login command.

   ```
   oc login -u kubeadmin -p <password>
   ```

2. Delete the aiops-ir-ui-api-graphql pod.

   ```
   oc delete pod -l component=aiops-ir-ui-api-graphql -n <cp4aiops_namespace>
   ```

3. Wait for the pod to restart.

Alerts tab shows "An unknown error occurred" error when all alerts are closed

If you are viewing a closed incident that has all associated alerts resolved, you can encounter an error when you view the Alerts tab. This "An unknown error occurred" error displays when there are no associated alerts. You can ignore this error message as the incident and alerts are resolved and closed.

In the metric anomaly details chart PNG and/or JPG export does not appear to work

This can occur intermittently, especially in the Firefox browser. The export time can be slower than normal so it could be that it takes a little longer for the export to complete. However, if nothing happens, try again later. Alternatively, use a different browser such as Google Chrome.

Same runbook status for multiple alerts in an incident

If more than one alert meets a runbook policy's conditions, the same runbook can be assigned to multiple alerts in an incident. From the incident overview page, you can select an alert and run an associated runbook. The runbook **Status** of the selected alert will be updated on the UI. However, the runbook status of other alerts in the incident might be updated with the same status. This is a known issue.

Closed incidents are missing details and displaying a critical error

Incidents with status of "Closed" are missing topology information on the incident **Overview** tab. The associated alerts of the closed incidents are also missing from **Alerts** tab. The following critical error is displayed on the **Topology** tab for closed incidents: "No resource exists with the specified identifier and time point".

Metric anomaly chart unexpectedly changes from zoom view to normal view

This can occur when a related alert is selected and added to a chart you are zoomed in on. The chart resets to normal view. To resolve, zoom in again after related alert is added to chart.

Unable to add metric anomaly in Related alerts to chart

In some cases, when you click the checkbox in the *Related alerts*, it does not add the related anomaly to the metric anomaly chart.

Limitation of preview text for default recommended action

When an alert is generated from any default log anomaly detection models, the preview of a recommended action might contain partial texts and does not reflect the full view of the recommended action. The first 4000 characters are extracted from the original resolution or action document webpage where possible, from which nonreadable texts such as URLs are excluded to form the content of the preview text.

Alert right-click menu items cannot be added if any actions have parameters with type 'array'

When automation actions exist that have any parameters of type 'array', the **Add menu item** button under menu configuration does not work as this type is not supported. The issue is limited to actions of type Ansible and Powershell which allow parameters of type 'array'.

To avoid or resolve this problem, identify actions with 'array' type parameters, and convert them to non-array parameters or delete them.

Metric search page chart lines are disjointed

In the Metric search page chart, the normalised forecast line is disjointed from the baseline data line. This is because the forecast data is normalised independently from the baseline data. Although the lines might not match up, the values shown in the tooltips are correct.

Policies

- Condition **Value** field changes `String:alert` to `Value of:alert`
- Breadcrumb navigation missing in policy editor
- Condition **Matches** field for numeric **Operator** selections
- **Last run** time and **Matched** count is updated in policies other than the trigger policy
- Double scroll bars on browser window
- Policy processing failing due to long policy conditions

- No ServiceNow ticket created by incident creation policy
- Datarouting pods can fail and need to be restarted
- Using default values in automatically run runbooks not working
- Netcool alert not suppressed when X in Y suppression policy conditions are met
- Cloud Pak for AIOps Impact policy fails to run when 'No input' is selected

Condition Values field changes "`String:`" to "`Value of:`"

For example, in a policy condition if the string "`alert.id`" is typed in the **Values** field, and then "`String:alert.id`" is selected, it is changed to "`Value of:alert.id`".

To prevent this, avoid a string that exactly matches the keyword. In this example, use the following condition instead:

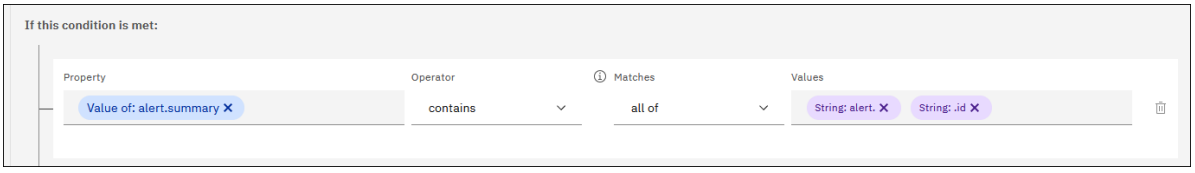


Figure. Policy condition

Note: This example is not an exact match of the string **alert.id**. This workaround finds all summaries containing **alert.** and **.id**.

Breadcrumb navigation missing in policy editor

In some cases where a policy name is long, the breadcrumb navigation in the top left of the policy edit session can be abbreviated. Clicking the breadcrumb still returns you to the Policy UI.

Condition "Matches" field for numeric "Operator" selections

When using a numeric **Operator** in a policy condition set (for example, **greater than**, **less than**, **greater or equal**) all options can be selected under **Matches**. However, always select **Only** for use with a numeric operator.

"Last run" time and "Matched" count updated in policies other than the trigger policy

In a case where an alert meets the incident-creation conditions of multiple policies, only one incident is created. However, all policies that proposed an incident, and the system incident creation policy has the same **Last run** times on the **Policies** hub. Each of these policies also increment their **Matched** counts by 1 in the **Details** tab of the side panel.

Double scroll bars on browser window

An extra scroll bar might appear on the right side of the browser window.

Policy processing failing due to long policy conditions

Policies that have conditions that are long (100,000+ characters) can cause policy processing to fail, resulting in no alerts or incidents being created. Failure can occur when temporal correlation training generates groups containing many alerts, and those alerts have long resource or type fields.

If this problem occurs, such policies should be disabled from the automation policy UI. Identify the policies by

1. Filtering for analytics policies.
2. Sorting by last run time (to identify those that have been processed recently and are likely triggering the problem).
3. Viewing the specification for each to see whether any have a long condition.

Any policies that show up by implementing these steps should be disabled.

No ServiceNow ticket created by incident creation policy

An alert meets the incident-creation conditions of a policy, but no ServiceNow ticket is associated to the incident.

To avoid this, complete the following steps under **actions** in the policy editor:

1. Click **Assign and notify**.
2. Select **In an existing channel**.

Datarouting pods can fail and need to be restarted

After installation, data for display in the UI, such as in the policy list, can be missing or stale.

If this issue occurs, restart the datarouting pods. You can identify these pods by using the following command:

```
oc get pods | grep datarouting.
```

Using default values in automatically run runbooks not working

This problem can occur by selecting a **Default parameter value** when creating a policy to assign runbooks to alerts. The **`useDefault`** value is not passed during automatic execution but is passed during manual execution.

You can execute the runbook manually from the **Runbooks** page.

Netcool alert not suppressed when X in Y suppression policy conditions are met

If you create an X in Y suppression policy that matches an alert originating from an IBM Tivoli Netcool/OMNIbus environment, the alert will not be suppressed.

Cloud Pak for AIOps Impact policy fails to run when 'No input' is selected

When enabling a Cloud Pak for AIOps Impact policy to Invoke an IBM Tivoli Netcool/Impact policy, the policy fails to run when **No input** is selected under the policy parameter mapping options. To avoid this, select one of the other mapping options. For example, select **Send the full alert**.

Secure Tunnel

- Secure Tunnel connector is not running after restart with Podman

Secure Tunnel connector is not running after a restart with Podman

This issue can occur when you install the Secure Tunnel connector to a host machine on which Podman is installed. When the host machine is rebooted and the Secure Tunnel connector is checked by using the **`podman ps -a`** command, the Secure Tunnel connector container does not display **`running`** status.

If this issue occurs, the **`podman-restart`** service must be activated by using the **`systemctl`** command:

```
systemctl start podman-restart
systemctl enable podman-restart
```

After entering the command, check **`podman-restart`** worked by using the following command:

```
systemctl status podman-restart
```

If the Connector is still not running, try restarting the host machine.

Runbook Automation

- When **`alert.suppressed`** value is used, runbook does not automatically run

When `alert.suppressed` value is used, runbook does not automatically run

Normally, you can select a runbook and configure it to run automatically: when an alert is converted to an incident, the runbook is assigned and runs automatically. However, if the parameter value **`alert.suppressed`** is used, the runbook does not run automatically as it reads this as a Boolean value rather than a string value. Therefore, it is necessary to manually run the runbook.

AIOps insights

- Number of Incidents reported in *Noise reduction* chart is inconsistent with number of Alerts and Events
- AIOps insights dashboard fails to load even when data is available
- Events not showing up on *Noise reduction* chart

Number of Incidents reported in Noise reduction chart is inconsistent with number of Alerts and Events

In the **AIOps insights** dashboard, the *Noise reduction* chart normally indicates the number of alerts, events, and incidents reported over a specific time period. However, the inclusion of historical data – containing longstanding, unresolved alerts – can result in a skewering of the data that is presented on the chart: the number of incidents that are presented can outnumber the number of alerts and events. Normally, this number is less than either – alerts reduce to a smaller number of events and events reduce to a smaller number of incidents.

The anomalous incident number happens because the reduction time frame covers alerts and events that are generated in the time period selected (for example, 7 Days). However, the incidents are generated from all outstanding alerts, including alerts that are not resolved, historically: alerts that occurred before the selected time period. So, in these circumstances, while the number of alerts and events is correct, the number of incidents is not.

AIOps insights dashboard fails to load even when data is available

Large amounts of data can cause the dashboard to fail to load or time out with the message **`Error – Metrics unavailable`** displayed for each chart. The problem is a scaling issue. The **AIOps insights** dashboard is not yet developed enough to handle huge amounts of data. A possible workaround is to increase resources for **`insights-api`** and **`elasticsearch`** pods. However, this approach might not be successful.

Events not showing up on *Noise reduction* chart

The charts in **AIOps insights** cover a timeline no greater than 30 days. The dashboard reads the **`firstOccurenceTime`** value from only within that period. If an alert was created outside of that timeline, and deduplicated, it is not added to the **`eventCount`** in the **AIOps Insights** Noise reduction chart. In this scenario, the **`eventCount`** for the alert increments in IBM Cloud Pak for AIOps, but not in the *Events* segment of the Noise reduction chart.

Ticketing

IBM Change Risk Assessment tab in ServiceNow not displaying change risk assessment details

In a proactive ChatOps channel, you can click the Change request ticket URL to view its details in a ServiceNow instance. However, in some cases, details might not be displayed in the **IBM Change Risk Assessment** tab.

Version 4.1.1 of the ServiceNow App requires specific values for the `x_ibm_waiops.admin` user that was used to create the Cloud Pak for AIOps ServiceNow integration.

1. If you have system admin access to the ServiceNow instance, go to **Organization**>**Users** and select the user who is assigned to the `x_ibm_waiops.admin` role.
2. Edit this user with the following information:
    - In the **User ID** field, enter `ibm.aiops`.
    - In the **First name** field, enter `IBM Cloud Pak for AIOps`.
    - Leave the **Last name** field empty.

After you update the user ID and name, update the Cloud Pak for AIOps ServiceNow integration to use the `ibm.aiops` user ID.

1. From the Cloud Pak for AIOps console, go to **Define**>**Integrations**.
2. Edit the existing ServiceNow integration or create an integration with the updated `ibm.aiops` ServiceNow user ID.

Change risk details now appear on future change risk assessments.

To view existing change risk details, click the **Notes** tab of a change request or go to **IBM Cloud Pak for AIOps**>**Change risk** and view the change risk table.

Deprecated and removed functions

Frequent updates to the product and changes in technology require that some function is deprecated or removed from support. See the following sections for definitions of stabilized, deprecated, and removed function and tables that identify items from these categories.

Deprecated functions

A *deprecated* item is supported, but no longer recommended for use and might become obsolete. Consider the alternative actions in the *Recommended action* and details that are provided in the following table:

Table. Deprecated function

Product or category	Affected item	Status	Version	Recommended action	More details and links
Microsoft System Center Virtual Machine Manager (SCVMM)	Infrastructure Automation - Infrastructure management	Deprecated	3.6.0	No change in version 3.6.0	Microsoft System Center Virtual Machine Manager (SCVMM) might be removed as a supported platform for Infrastructure management in a future release.
Advanced Content Runtime	Infrastructure Automation - Managed services	Deprecated	3.3.0	No change in version 3.6.0	Advanced Content Runtime might be removed in a future release. If removed from Infrastructure management, you will need to import Advanced Content Runtime templates yourself.

Product or category	Affected item	Status	Version	Recommended action	More details and links
PostgreSQL 10 (Infrastructure Automation)	Infrastructure Automation - Infrastructure management	Deprecated	3.6.0	Upgrade to PostgreSQL 13	• If you need to upgrade for appliances, see Migrating Red Hat CloudForms and IBM Cloud Pak for Multicloud Management appliances to Infrastructure Automation • If need to upgrade for a podified installation, see Upgrading PostgreSQL for a podified Infrastructure management installation.

Removed functions

A *removed* item is typically function that was deprecated in previous releases and is no longer available in the product. You must use alternative features as a replacement for the removed function. Consider the alternative actions in the *Replacement* and details that are provided in the following table:

Table. Removed function

Product or category	Affected item	Version	Replacement	More details and links
Kubernetes connector and documentation	IBM Cloud Pak for AIOps	4.1.0	Configure a Kubernetes observer job as a replacement.	
VMWare connector and documentation	IBM Cloud Pak for AIOps	4.1.0	Configure a VMware NSX observer job or VMware vCenter observer job as a replacement.	
Red Hat® OpenShift® Container Platform 4.8	IBM Cloud Pak for AIOps	3.7.1	Use Red Hat OpenShift Container Platform 4.10 or 4.12	Supported Red Hat OpenShift Container Platform versions
Red Hat OpenShift Container Platform 4.10	IBM Cloud Pak for AIOps	4.2.0	Use Red Hat OpenShift Container Platform 4.12	Supported Red Hat OpenShift Container Platform versions
GitHub connector	Data tool and connections	3.6.0	--	--
IBM Cloud Pak CLI (`cloudctl`)	IBM Cloud Pak for AIOps, for offline install and upgrade	3.6.0	Use the `ibm-pak` plugin for the OpenShift `oc` CLI tool instead of `cloudctl` commands.	Install and upgrade instructions for offline deployments now use `oc ibm-pak` commands instead of `cloudctl` commands.
Event Manager Gateway	IBM Cloud Pak for AIOps and IBM Netcool Operations Insight deployment scenario	3.6.0	Use the Netcool connector	Installing IBM Cloud Pak for AIOps with IBM Netcool Operations Insight or on-premises Netcool Operations Insight in version 3.6.0

Product or category	Affected item	Version	Replacement	More details and links
Log display support	Infrastructure management	3.6.0	--	Log display support on the UI is removed
Terraform 0.11.15	Infrastructure Automation - Managed services	3.6.0	--	You might be able to still use Terraform 0.11.15, but you will need to upload the required engine, providers, or content yourself.
Terraform 0.12.31	Infrastructure Automation - Managed services	Deprecated	3.7.0	No change in version 3.7.0
Terraform 0.13.7	Infrastructure Automation - Managed services	Deprecated	3.7.0	No change in version 3.7.0
Terraform 0.14.11	Infrastructure Automation - Managed services	Deprecated	3.7.0	No change in version 3.7.0
Terraform 1.1.9	Infrastructure Automation - Managed services	Deprecated	3.7.0	No change in version 3.7.0
New installs - certificates provided by IBM Automation foundation	IBM Cloud Pak for AIOps and Infrastructure Automation	4.1.0	Red Hat OpenShift Container Platform cluster certificates or user created custom certificates	What's new - install and upgrade in version 4.1.0
User interface using a different storage class	IBM Cloud Pak for AIOps	4.1.0		

Overview of IBM Cloud Pak for AIOps

IBM Cloud Pak® for AIOps is an AIOps platform that deploys advanced, explainable AI using the IT Operations (ITOps) toolchain data so that you can confidently assess, diagnose, and resolve incidents across mission-critical workloads.

IBM Cloud Pak for AIOps eases the path to adopting advanced AI for ITOps to decrease your operational costs. With this Cloud Pak, you can increase your customer satisfaction by proactively avoiding incidents and accelerating your time to resolution.

The scale of IT systems and their complexity is continually increasing over the last few years because of digital transformation, containerization, and hybrid cloud adoption. IT teams are being inundated with routine maintenance activities and expanding cloud services, leaving them little or no time to contribute toward innovation. To accelerate business automation, reduce complexity, save costs, and automate regular tasks, companies must use the power of AI.

IBM Cloud Pak for AIOps powers automation by using diverse data sets from an entire range of hybrid environments from cloud to on-premises, and bringing the information together across ITOps. With this Cloud Pak, you can tap into shared automation services to get insight into how your processes run. You can also visualize hotspots and bottlenecks, and pinpoint what to fix with event detection to prioritize which issues to address first.

IBM Cloud Pak for AIOps helps you uncover hidden insights from multiple sources of data, such as logs, metrics, and events. The Cloud Pak delivers those insights directly into the tools that your teams already use, such as Slack or Microsoft Teams, in near real-time. Included AI management tools provides you with unprecedented visibility into your organization's infrastructure so that you can predict failures and facilitate problem resolution.

For information about the IBM Cloud Pak for AIOps architecture, see Architecture.

Concepts

IBM Cloud Pak for AIOps represents the state of managed entities by using the following three concepts:

- Events, which are a record containing structured data that summarizes key attributes of an occurrence on a managed entity. Events indicate a change in the state of a service or a system. Events are a category that includes the raw data, alerts, and incidents.
- Alerts, which represent an ongoing anomalous condition against a single managed entity. An alert might evolve over time as the condition changes, and extra context, and insights can be associated with it. An alert presents a single symptom, typically composed of one of more signals, which are immutable, and point-in-time statements about the entity in question.

  For example, an alert which indicates that a given resource is using 100% of its available storage is made up of multiple incoming signals of the disk usage at different times. The alert represents the ongoing time span that the resource in question is in this state.
- Incidents, which represent the context around an issue currently impacting operations. An incident is a collection of insights that come from different data sources, such as events, topology, and logs. Incidents bring together key elements to provide a holistic view of the problem. These elements include alerts, information about how the affected resources are related, guidance on possible resolutions, and automations to remediate the issue. A key difference from an alert is that while an alert has context and insights on a single entity, and a single symptom (alert type), an incident brings together all the symptoms for a more comprehensive understanding. The creation and evolution of incidents are informed by alerts.

  For example, an incident might bring together alerts on a full disk, alerts on increased request failures, and alerts on application failures. These lists of alerts can be paired with guidance on how to address each separate problem and automations, which can perform the solutions.

Features and benefits

With IBM Cloud Pak for AIOps, you can accelerate your journey toward AI powered operations. You can use the Cloud Pak to set up an integrated, AI-driven set of tools to pull, and analyze data from multiple IT platforms to help with ITOps management. IBM Cloud Pak for AIOps combines a set of capabilities to provide a single solution that facilitates predicting, communicating, and resolving events before they become serious problems. Issues are inevitable in any IT landscape, it's how and when you respond that can make a world of difference to your organization.

IBM Cloud Pak for AIOps includes the following key features:

- Guided tours
- Connectors and data ingestion
- AI modeling and training
- Resource management
- Topology
- Log anomaly detection
- Metric anomaly
- Event grouping
- Similar ticket comparison
- Incidents and alerts
- ChatOps
- Automations
- Change risk
- Additional capabilities
  - Infrastructure automation

These capabilities are supported by a network of connectors and capabilities that manage all facets of the AIOps lifecycle from model training to execution.

IBM Cloud Pak for AIOps collects information from all of your IT assets, such as applications, the infrastructure that they run on, and the networking systems that support them. It then uses that data to uncover hidden insights and identify root causes of events. Using this data to train your models lead to better event discovery and a more accurate understanding of your topology. With this advanced understanding of your topology, you can pinpoint where events occur and how far-reaching their impact is. When IBM Cloud Pak for AIOps detects potential incidents, it creates an incident about the incident. Then, it uses a

ChatOps environment, such as Slack or Microsoft Teams, to notify, update, and provide your team with near real-time potential remedies, which empowers your teams to automate tasks with clear and open AI decision-making.

The user experience is simplified around AI lifecycle management. The modern experience breaks the complex task of AI model training into a simpler milestone-based experience. This makes it easier for any user to select, filter data, train the models, and view the progress and its output. Through this new experience, you can train the built-in AI models of change risk, event grouping, log anomaly detection, and similar tickets with your data with just a few clicks.

Guided tours

First-time users are presented with a guided tour upon login. This tour is customized for IBM Cloud Pak for AIOps to introduce users to the various home page components, shows how to access tools and capabilities, and where to complete key tasks across the platform.

The home page content is customized based on the user's role-based access. The initial tour shows the user the following components:

- Insights-related summary cards that provide a high-level glance at important metrics in the environment. These insights are related to only the components that the user has access to view or manage.
- The *Quick navigation* and *Support* links for easily accessing commonly used tools, product documentation, IBM Support, and more.
- The main navigation menu for the IBM Cloud Pak Automation console, which lists the tools and pages that are accessible by the user.
- The *Tours* menu for accessing more tours and guidance on how to use the Automation console and completing essential tasks. The tour menu includes links for the welcome tour, and for tours to help get started with data and tool connections and AI modeling.
- The getting started links for setting up connections, learning more about IBM Cloud Pak, and managing user access control.

More tours can be accessed at any time by the user to get a better understanding of how to get started with different tasks and components. Users can select a tour by clicking the **Tours** icon in the Automation console toolbar, and clicking their preferred tour. The following overview tours are available for IBM Cloud Pak 4.2.0:

- **Welcome to IBM Cloud Pak for AIOps**

  A repeat of the initial *Welcome* guided tour.

- **Discover AIOps insights**

  Overview of the *AIOps insights* dashboard, which summarizes how IBM Cloud Pak for AIOps helps organizations anticipate, troubleshoot, and resolve IT incidents. The tour loads sample data to walk the user through available toolbars and charts, including Mean time to restore, Noise reduction, Incident activity, Runbook usage, and the optional, nonproduction use Applications disruption costs chart.

- **Start with data and tool connections**

  Available from the Data and tool connections and the home page. This tour guides users through the Data and tool connections to show how to add connections between IBM Cloud Pak for AIOps and other tools.

- **Introduction to resource management**

  Available from the *Resource management* tool. This tour identifies the resource management capabilities, which you can use to gain an overview of applications, resource groups, and resources and select them to view their details and topology. The tour introduces you to resource groups, which you can use to build application definitions. These resource groups are created automatically when you connect tools, however, you can add groups by using the template page.

- **Understand applications and resource groups**

  Available from the *Resource management* tool when a user selects an application or a resource group to view details. This tour shows a user how to view incident details, view affected resources, view and navigate the timeline, and how to get insights from changes to a resource over the time.

- **Learn about the timeline**

Available from the *Resource management* tool. This tour is available when a user selects to view details for an application that has an associated incident. This tour walks the user through the details of the timeline widget, showing how to interact with the data and see changes that have occurred for alerts.

- **Explore AI modeling**

  Available from the AI Model Management and the home page. This tour helps you to understand the two types of AI algorithms that are available to enable AI modeling. This tour uses the *Change risk model* summary card to show you how to set up and monitor the training of model-generating algorithms.

- **Introduction to automations**

  Available from the *Automations* tool, which the user can use to proactively set up actions and policies to detect and remediate events, both current and future. With this tour, a user learns how to create policies, view policy details, create and view runbooks, and more.

- **Create a new policy**

  This step-by-step tutorial is available from the *Automations* tool to guide users in the creation of a new policy. This step-by-step tutorial uses the *Promote alerts to an incident* policy template to help you understand the basics of setting up a new automation policy.

- **Introduction to incidents and alerts**

  Available from the *Incidents and alerts* tool. This tour provides an overview of how incidents bring insights together to present the holistic view of an incident. The tour guides you through the different tabs, options, and views of the *Incidents and alerts* tool to show you how to view your existing incidents and alerts, including how to filter the incident list, drill down to view alert details, clear alerts, and more.

- **Investigate an incident**

  Available from the *Incidents and alerts* tool when a user views an incident in detail. This tour guides a user through the process to investigate an incident. A user is introduced to how to view probable causes and top alerts, review associated automations, change incident settings, investigate incident alerts, and how to visualize an incident. The tour also introduces details on incidents, trigger alerts, and the legend and timeline for visualizing an incident.

Connectors and data ingestion

Data is the life blood of AI but configuring data feeds can be complex. You can use the Automation console tools and connectors to set up connections to a wide variety of systems for data ingestion. IBM Cloud Pak for AIOps supports connections to large system solutions including log management, event management, and service management tools within an IT environment, such as Splunk, Falcon LogScale, LogDNA, PagerDuty, and ServiceNow. You can also use the Netcool probes to ingest events into IBM Cloud Pak for AIOps.

A gRPC framework is used for some connectors to support local or remote deployment of a bidirectional data flow. These connectors, such as the Instana, Slack, and GitHub connector, can be deployed locally in the same project (namespace) as IBM Cloud Pak for AIOps, or deployed remotely.

For the full list of supported connections, see Configuring data and tool connections.

Netcool integrations

Connectors are available for connecting to IBM Tivoli Netcool/OMNIbus and IBM Tivoli Netcool/Impact.

A *Netcool* connection is used to define a connection to IBM Tivoli Netcool/OMNIbus. An IBM Tivoli Netcool/Impact connection is used to define a connection to IBM Tivoli Netcool/Impact. For more information about these connections and integrating with Netcool, see Netcool integrations.

Instana connections

An Instana connector is available for connecting to and integrating with Instana to enable connectivity to Instana events, metrics, and topology. Instana assets can now be combined with IBM Cloud Pak for AIOps AI, such as runbooks, metric anomaly detection, and log anomaly detection to help you gain insights. For more information about setting up this connection, see Creating Instana connections.

Ansible connections

IBM Cloud Pak for AIOps also includes Ansible connectors for connecting to Ansible and an SSH connector to connect to another system where scripts and commands can run.

The Ansible connection provides you with the ability to set up a connection to an Ansible automation platform controller server to gain benefits from the configuration and orchestration capabilities of Ansible. For more information about setting up the connection, see Ansible automation provider.

The SSH connection is agentless and connects directly to the target machine. It authenticates by using SSH public key authentication. You can only create one SSH Connection. For more information, Creating SSH connections.

AI modeling

You can train AI models to hone its ability to derive insights from your incoming data connections. To get the most out of your AI models, you can manually map your log data to the JSON training format and train your models with it. For more information about training specific types of models, including suggested mappings for events and logs, see the following training topics:

- About AI algorithms
- Configuring AI models

For more information about what you can do to gain further insight into your IT infrastructure, see the following topics:

- Navigating the ChatOps interface
- Managing incident

For guidance on how to plan data connection and AI model training, see the following topic: Planning data loading and training.

Resource management

IBM Cloud Pak for AIOps brings together the capability to group resources from different data types into applications (services). You can flexibly define an application (service) to meet your business needs. With applications (services), you can obtain an integrated view of your resources to understand inter-dependencies.

IBM Cloud Pak for AIOps focuses on the user experience and health of your applications (services). This application-centric approach uses a built-in application template model to dynamically aggregate and group tagged resources to enable rapid visualization of different application components.

Topology

An integrated service and topology management and topology viewer provides complete up-to-date visibility of your applications and resources, and provides you with control over dynamic infrastructure and services. You can configure your topology visualizations to also show real-time or historical data.

In addition, to help you pinpoint incidents within applications and resource topologies, IBM Cloud Pak for AIOps tools work together to correlate events.

Log anomaly detection

IBM Cloud Pak for AIOps includes a pair of AI algorithms to gather log data from one or multiple components in the application architecture to identify a baseline of expected log message types. IBM Cloud Pak for AIOps uses this baseline to discover abnormal behavior in your live log data. These AI algorithms can run independently of the other. If both algorithms are enabled, then any log anomalies discovered by both are reconciled, so that only one alert is generated. In this case, the severity of the combined alert is equal to the highest severity of the two alerts. Each algorithm is used to gather data differently:

- Natural language log anomaly detection uses natural language techniques on a subset of your log data to discover abnormal behavior. This log anomaly detection takes large amounts of log data and trains on it to learn what is normal behavior for a particular component. This model goes beyond just looking at error states or frequency of metadata around log messages. Instead, it determines when something becomes an anomaly compared to what patterns it typically exhibits.

- Statistical baseline log anomaly detection uses a statistical moving average on all of your log data to discover abnormal behavior. This algorithm automatically detects unusual patterns in logs and notifies you when they occur. Data that is used for analysis is updated every 30 minutes so this algorithm provides value quickly.

For more information, see Log anomaly training definition.

Metric anomaly

Metric anomaly detection provides the ability to analyze any time series metric automatically using AI without the need for configuration or tuning. This AI leverages the Instana Integration, automations, and policies. You can group metric anomalies with log anomalies and other Alerts.

Metric anomaly detection can help you measure key performance indicators, like memory availability or latency, to identify trends and deviations in these areas early.

For more information, see About metric anomaly detection.

Event grouping

AI algorithms are provided with IBM Cloud Pak for AIOps to group events together and present the events within a single incident.

- Temporal grouping is used to group events that are discovered to co-occur over time. When a problem arises, typically multiple parts of a system or environment are impacted. When events in different areas co-occur, it makes sense to look at them together and treat them as one problem to try to determine what might have occurred. Grouping co-occurring events together reduces the number of tickets and incidents that are opened and the number of people looking at the same problem, thereby significantly reducing noise in your monitoring systems. It helps you to understand the context of an issue so you can prioritize, triage, and resolve it more quickly.
- Topological grouping is used to group events based on the resource groups in which events occur. For example, if you have a resource group that is made up of all the resources within a given Kubernetes namespace, then any events on pods, microservices, or other resources in that namespace are grouped together in a single topological group. Topological grouping helps you understand when events are connected based on their topology, providing valuable context for why related events might occur together.
- Scope-based grouping automatically groups events relating to an incident if they have the same defined scope and occur during the same period. A scope can be used to identify where events originate based on a common attribute, for example, the location of a server room. By understanding when events are related based on both time and location, you can more quickly diagnose incidents.

For more information, see Event grouping training definition.

Similar ticket comparison

The similar ticket AI algorithm in IBM Cloud Pak for AIOps can be used to define an AI model so that when incidents occur, the model can review details for similar tickets from the past to help determine a resolution to current problems. This model aggregates information from past similar tickets. It can also extract the steps that are used to fix previous incidents, if documented. This AI algorithm discovers details about similar messages, anomalies, and events within your tickets that occurred in the past and are impacting the current application. For more information, see Setting up training for similar tickets.

Incidents and alerts

IBM Cloud Pak for AIOps enables you to identify health issues across your application, services, and network infrastructure on a single management console. It provides a Incidents and alerts tool that brings together incidents, alert details, alert severity, and alert filtering. This capability helps you to analyze the context of incidents across the full enterprise estate. This viewer acts as a single pane of glass incident dashboard, which displays key details like the probable cause, topology, and associated automations to resolve the incident.

Some alerts in the console are presented in alert groups, based on analytics. By using the alert list, you can explore why these alerts were grouped together, and this understanding can further help with alert resolution.

Alerts are displayed in the Alert Viewer. From a ChatOps integration (Slack or Microsoft Teams), you can launch into an Alert Viewer that shows the alerts that are associated to a specific incident.

For more information, see Managing incidents and alerts.

ChatOps

IBM Cloud Pak for AIOps ChatOps support includes connections with existing social collaboration tools, Slack, and Microsoft Teams. These connections help to deliver AI-powered automation for ITOps and management within their preferred collaboration tool of choice.

From a ChatOps integration you can launch into the Automation console Incidents and alerts tool to view more information about an alert that is being viewed in ChatOps.

IBM Cloud Pak for AIOps uses ChatOps to provide a message-based interface for reporting incidents with the ITOps that you are monitoring. This ChatOps interface displays the following types of operations information:

Table 1. Operations information available from the ChatOps interface

Operation	Description
Anomaly Detection	Provides details about detected anomalies in your real-time or offline data, such as the time when the incident was detected, the incident description, status, and more.
Relevant events	Groups-related events to aid incident diagnosis. Events can include alerts, PagerDuty alerts, or log anomalies.
Probable cause	Derives root fault component, and derives the full scope of components that are affected by an incident.
Ticket Similarity	For a particular ticket, finds the highest "n"-ranked similar tickets from the past.
Next Best Action	For a particular incident, suggests the highest "n" actions from similar tickets from the past.

Automations

with IBM Cloud Pak for AIOps an engine and UI for automation is included that you can use to create automation policies, runbooks, and actions to help you resolve incidents quickly by setting up and enabling an automatic response as situations arise. Starting up additional capacity or restarting a failed service can happen before your team knows there is an issue.

For setting up automated procedures, you can proactively set up policies, runbooks, and actions to detect and remediate events, both current and future.

- Policies are rules that contain multiple condition and action sets. They can be triggered to automatically promote events to alerts, reduce noise by grouping alerts into an incident, and assign runbooks to remediate alerts. For more information, see Policies.
- Runbooks automate procedures that do not require human interaction, thereby increasing the efficiency of IT operations processes. With runbooks, your operators can spend more time innovating and are freed from performing time-consuming manual tasks. For more information, see Runbook Automation.
- Actions in runbooks are the collection of manual steps grouped into a single automated entity. An action improves runbook efficiency by automatically performing procedures and operations. For more information, see Actions

Change risk

IBM Cloud Pak for AIOps includes an AI algorithm for providing change risk assessment and includes the capability of surfacing change risk assessments through ChatOps.

The change risk AI algorithm provides an assessment of the risk of implementing a proposed change; for example, a code change or a new software version. Hundreds of changes can affect an application during its lifecycle. Change risk correlates historical change data with historical incident data and determines how likely it is that a specific code change would cause a problem. Using the assessment percentage provided, you can determine how safe it is to proceed with the change. Training this AI model can help you ensure that risky changes for an application are assessed before deployment. For more information, see Change risk training definition.

The change risk assessment in ChatOps provides you with a model confidence percentage that represents the confidence in the AI's prediction of the impact of a change. This percentage is relative to the risk associated with a new or updated ServiceNow change request ticket. The model confidence percentage is calculated based on the change risk AI model that is based on the change risk AI algorithm and the data that you feed into it. Change risk can be evaluated based on any type of change, whether it is changes to code, or to configurations, or to data. If you have historical change request tickets for that type of change, IBM Cloud Pak for AIOps can provide a model confidence percentage for it. For more information, see Change risk in ChatOps.

Infrastructure automation

Infrastructure automation is included as an optional feature with IBM Cloud Pak for AIOps. This Infrastructure automation component consists of the following components, which were available with IBM Cloud Pak for Multicloud Management:

- Infrastructure management, previously called IBM Red Hat CloudForms.
- Managed services, previously called Terraform & Service Automation or IBM Cloud Automation Manager.

Infrastructure management delivers the insight, control, and automation enterprises need to address the challenges of managing virtual environments, which are far more complex than physical ones. This technology enables enterprises with existing virtual infrastructures to improve visibility and control, and those just starting virtualization deployments to build and operate a well-managed virtual infrastructure.

Managed services provide you with the capability to automate provisioning of infrastructure and virtual machine applications across multiple cloud environments with optional workflow orchestration.

Infrastructure automation enables IT Operations and Site Reliability Engineer (SRE) teams to use infrastructure as code practices to drive IT velocity and shift to the left of operations.

Infrastructure automation is installed within a separate cluster from IBM Cloud Pak for AIOps. For more information, see Infrastructure automation.

Bringing the features together

By combining AI-driven insights with a modern user experience, you can achieve reduced Mean Time to Resolution (MTTR) when an incident occurs and avoid incidents in the first place through a change risk advisor. The user experience brings in different types of data together and builds out an application perspective, delivering unique insights from the AI models.

Application impact avoidance

The trained AI models establish a baseline and automatically detect anomalies across structured and unstructured data. The models reduce noise and bring related entities across different data sets together into a complete “incident” ahead of a potential incident or application impact. A much smaller set of high-impact “incidents” versus a flood of events and incidents, provides deep insights with explainability.

These incidents help you to visually understand the probable cause to help prevent future downstream impact. The Next Best Action recommendation from past incidents, and the built-in Runbook Automation capabilities accelerate the time to remediation of an incident. The probable cause capability allows users to isolate and find the root cause of the problem faster.

Incident diagnosis

- Derive root fault component and probable cause of affected components to isolate problems faster.
- Detect anomalies by using automatic parsing to detect anomalies in logs and alerts earlier than rule-based alerts.

Insight delivery

- Integrate with pre-trained AI models to gain new insights from existing tools. Keep teams focused, surfacing insights and recommendations into existing workflows.
- Avoid application impact through correlating data across all relevant data sources. Detect hidden anomalies, anticipate issues, and resolve faster. Correlate a vast amount of unstructured and structured data in real time with AIOps tools.
- Proactively avoid risks and automate runbooks for more efficient workflows.
- Proactively avoid incidents with explainable AI that advises based on inputs across ITOps.
- Create a visual representation of the data that is flowing through your business and pinpoint bottlenecks in your operations.
- Draw on advanced AI and analytics to create a baseline that you’ll use to track key performance indicators (KPIs) over time.

Architecture

Built on Red Hat® OpenShift® Container Platform, IBM Cloud Pak® for AIOps combines a set of capabilities to provide a single solution that facilitates predicting, communicating, and resolving events before they become serious problems.

IBM Cloud Pak for AIOps integrates infrastructure and operations management into a single coherent structure across business applications, virtualized servers, network devices and protocols, and security and storage devices. IBM Cloud Pak for AIOps enables CIOs, site reliability engineers (SREs) and IT Operators by uncovering hidden insights from multiple sources of data (like logs, metrics, and events). It delivers those insights directly in to the tools that teams work with (like Slack, Microsoft Teams, and ServiceNow) in near real-time, and also presents them in a web-based user interface.

The core functions of IBM Cloud Pak for AIOps can be categorized into five capabilities: event management, incident detection and resolution, entity extraction, and insight delivery. These capabilities are supported by a network of connections that gather data. It also offers all facets of the AIOps lifecycle from model training to execution. These capabilities are bolstered by IBM Cloud Pak foundational services.

For information about the possible deployment configurations, see Installing.

Architectural diagrams

The following diagrams illustrate how these capabilities come together in IBM Cloud Pak for AIOps:

IBM Cloud Pak for AIOps

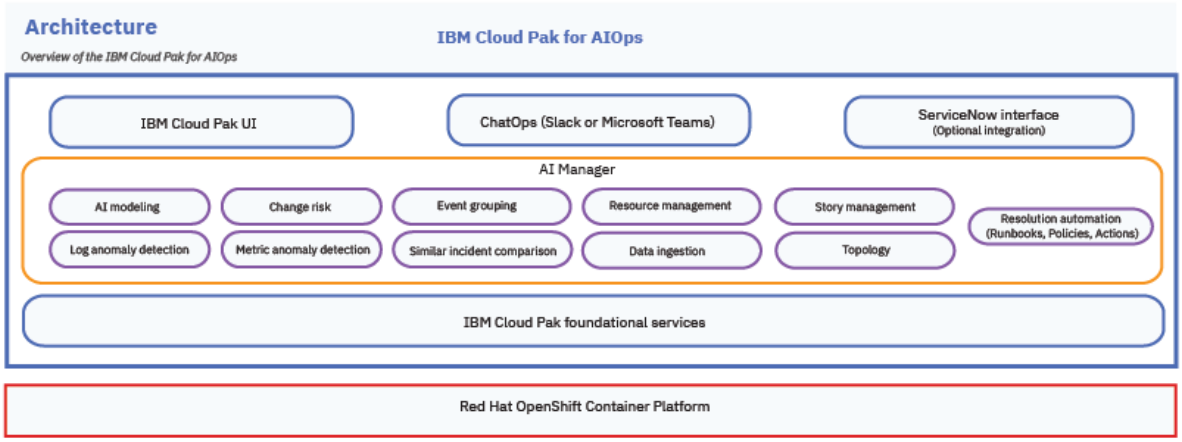


Figure 1. Architecture

IBM Cloud Pak for AIOps is installed on Red Hat® OpenShift® Container Platform. For more information about the supported versions of Red Hat OpenShift Container Platform, see Supported Red Hat OpenShift Container Platform versions.

When installed, IBM Cloud Pak for AIOps includes access to the IBM Cloud Pak Automation console. This UI is a web-based application for accessing features and capabilities for IBM Cloud Paks, such as IBM Cloud Pak for AIOps. If you have multiple IBM Cloud Paks installed in the same environment, you can switch between working with each IBM Cloud Pak from this single UI.

Installing IBM Cloud Pak for AIOps gives you access to the following features and capabilities:

- AI modeling
- Applications
- Change risk
- Data ingestion
- Event grouping
- Log anomaly detection
- Metric anomaly detection
- Resolution automation (Runbooks, Policies, Actions)
- Similar incident comparison
- Incident management
- Topology

For more information about these features and capabilities, see features and capabilities.

Infrastructure Automation

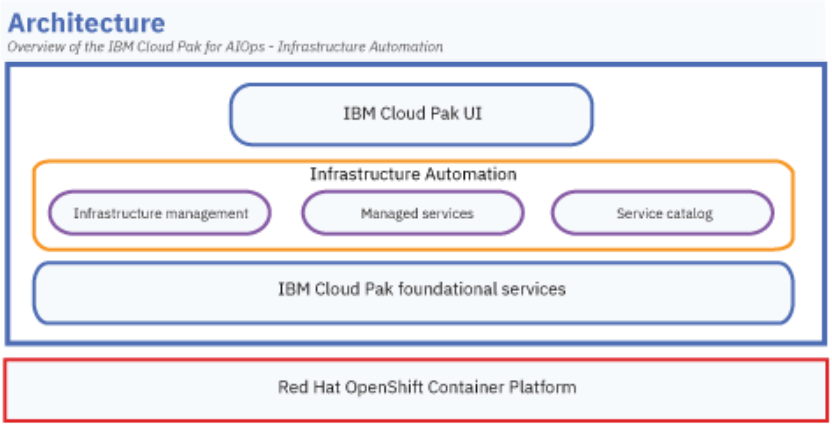


Figure 2. Infrastructure Automation architecture

Infrastructure automation is a standalone module within IBM Cloud Pak for AIOps. When you have a license for IBM Cloud Pak for AIOps, you are entitled to install and use Infrastructure Automation. Infrastructure Automation consists of these components (formerly part of IBM Cloud Pak for Multicloud Management):

- Managed services and service catalog, previously called Terraform & Service Automation or IBM Cloud Automation Manager
- Infrastructure management, previously called IBM Red Hat CloudForms.

Managed services provide the capability to automate provisioning of infrastructure and virtual machine applications using Terraform and Ansible. It includes supported versions of Terraform that works across multiple cloud environments with comprehensive IT workflow orchestration. Using out-of-box integration with Ansible, workflows can interconnect Terraform provisioning templates with Ansible playbooks for configuration management, fully supporting infrastructure-as-code (IaC) and GitOps best practices.The service catalog allows authors administrators to publish services and end users to self-service provision infrastructure and applications.

Infrastructure management delivers the insight, control, compliance and automation enterprises need to address the challenges of managing hybrid cloud environments, which are far more complex than physical ones. This technology enables enterprises with existing virtual, baremetal and cloud infrastructures to improve visibility, control and compliance, and those just starting hybrid cloud deployments to build and operate a well-managed hybrid cloud infrastructure.

Infrastructure Automation enables IT Operations and Site Reliability Engineer (SRE) teams to use infrastructure as code practices to drive IT velocity and shift to the left of operations. With modernize infrastructure management, you can perform the following tasks:

- Shorten the time to provision infrastructure (VMs, Cloud Services, Kubernetes) across on-premises, private, and public clouds
- Create repeatable templates for common infrastructure requests
- Enable self-service catalogs to avoid manual responses to tickets
- Scale infrastructure automatically based on demand
- Support day-2 operations including automated compliance
- You can provision, run, and manage workloads on the infrastructure of your choice.

IBM Cloud Pak for AIOps Infrastructure Automation is installed on Red Hat® OpenShift® Container Platform. For more information about the supported versions of Red Hat OpenShift Container Platform, see Supported Red Hat OpenShift Container Platform versions.

IBM Cloud Pak for AIOps with Infrastructure Automation

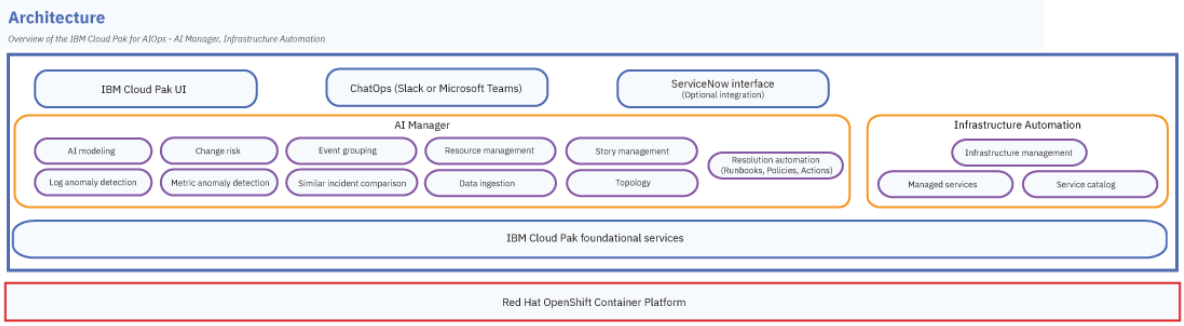


Figure 3. IBM Cloud Pak for AIOps with Infrastructure Automation architecture

Infrastructure automation can be used together with IBM Cloud Pak for AIOps to support use cases which require Terraform automation and service orchestration as part of the incident resolution process. Day 2 operations like changing the size of VMs, power management,or applying a patch are supported by Infrastructure Automation and can easily be accessed through a common console. They can also be automated through runbook actions by invoking Infrastructure Automation APIs. Both Infrastructure Automation and IBM Cloud Pak for AIOps integrate with ServiceNow and can be used together to calculate the risk score of infrastructure change requests.

Compliance

IBM Cloud Pak® for AIOps complies with the following security and privacy assessments, which can apply to all add-on services on IBM Cloud Pak for AIOps.

Customers are responsible for ensuring their own readiness for the laws and regulations that apply to them.

Customers are responsible for identifying and interpreting any relevant laws and regulations that might affect their users. Customers are also responsible for any actions their users might need to take to comply with these laws and regulations.

Customers must track whether personal information is present in their data. IBM Cloud Pak for AIOps is not aware of the presence of personal information in data imported from the client, such as log files. For more information about types of personal data that is processed by IBM Cloud Pak for AIOps, see Data Collection.

FIPS compliance

Federal Information Processing Standards (FIPS) are standards and guidelines that are issued by the National Institute of Standards and Technology (NIST) for federal government computer systems.

For more information, see Federal Information Processing Standards (FIPS).

FISMA compliance

For Federal Information Security Management Act (FISMA) level deployments, you need to follow the **FIPS compliance** and **Audit logging** sections.

For more information about FIPS compliance, see FIPS compliance.

For more information about Audit logging, see Audit logging.

For more questions around FISMA ready, contact your {{site.data.keyword.sales_rep_bp}}.

GDPR

Customers are responsible for ensuring their own readiness for the laws and regulations, including the European Union General Data Protection Regulation.

For more information, see the following topics:

- Security and Privacy by Design (SPbD)
- Platform considerations for GDPR readiness

Federal Information Processing Standards (FIPS)

Federal Information Processing Standards (FIPS) are standards and guidelines that are issued by the National Institute of Standards and Technology (NIST) for federal government computer systems.

The standards are developed when compelling federal government requirements for standards, such as for security and interoperability, exist, but acceptable industry standards or solutions do not exist. Government agencies and financial institutions use these standards to ensure that products conform to specified security requirements.

Encryption with FIPS support enabled

When FIPS support is enabled, IBM Cloud Pak for AIOps uses cryptographic modules that are compliant with Level 1 of the Federal Information Processing Standard FIPS-140-2. Certificates that are used internally are encrypted by using FIPS-approved cryptography algorithms. FIPS-approved modules can optionally be used for the transmission of data. Traffic inside the IBM Cloud Pak for AIOps boundary is still secure, as traffic between nodes is automatically encrypted at the Red Hat® OpenShift® Container Platform level when TLS protection is enabled, while traffic inside a given node happens in-memory and does not leave the node.

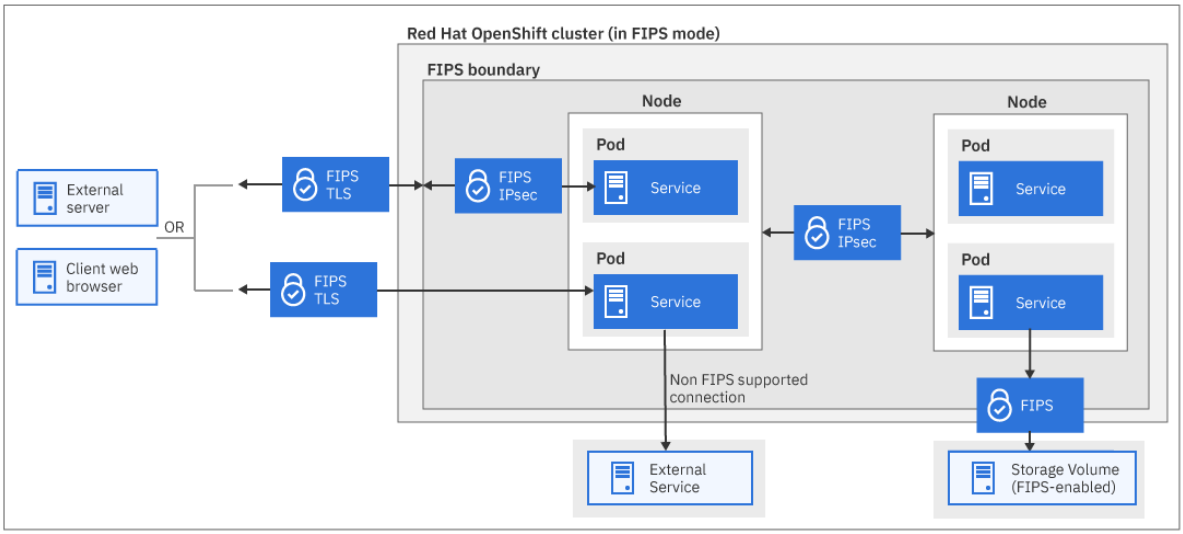


Figure. FIPs overview

FIPS (Federal Information Processing Standards) compliant encryption is validated for IBM Cloud Pak for AIOps services and components, including the IBM Cloud Pak foundational services that are used by IBM Cloud Pak for AIOps.

With FIPS enabled Data is FIPS encrypted at rest and inbound communications are FIPS encrypted. Outbound communications can support both FIPS enabled and non-enabled connections. For FIPS enabled connections, outbound connections rely on the server to ensure FIPS ciphers are chosen. To ensure that connections, including Observers, are FIPS enabled, an external service to mandate the use of FIPS compliant ciphers when negotiating encryption is required.

Enabling FIPS support

To enable FIPS support, you must enable this support when you are installing Red Hat OpenShift Container Platform and IBM Cloud Pak for AIOps. For more information, see:

- IBM Cloud Pak for AIOps: Enabling Federal Information Processing Standards (FIPS) support
- Infrastructure Automation: Enabling Federal Information Processing Standards (FIPS) support

Enabling Federal Information Processing Standards (FIPS) support for IBM Cloud Pak for AIOps

Learn how to enable FIPS and run IBM Cloud Pak for AIOps on a FIPS-compliant system.

To enable support you need to install IBM Cloud Pak for AIOps on a Red Hat OpenShift Container Platform cluster that is running in a FIPS enabled mode. You also need to complete some tasks before you begin installing IBM Cloud Pak for AIOps on your cluster and complete other tasks while you are installing IBM Cloud Pak for AIOps. Depending on whether you plan to enable or set up some optional connections, you might need to further complete tasks after your finish installing to ensure your connections support FIPS.

Important: You can only enable FIPS support when you are installing. You cannot upgrade a non-FIPS enabled environment to a FIPS enabled environment. You also cannot backup an IBM Cloud Pak for AIOps from a non-FIPS enabled environment and restore it into a FIPS enabled environment. The remote kafka connector feature is not be available on FIPS-compliant environments.

To enable FIPS support, complete the following tasks during your overall cluster and IBM Cloud Pak for AIOps installation:

1. Installing Red Hat OpenShift Container Platform and storage tasks required for compliance.

   These tasks must be completed when you are installing your Red Hat OpenShift Container Platform cluster and configuring your storage before you begin to install IBM Cloud Pak for AIOps on your OpenShift cluster.

2. Installing IBM Cloud Pak for AIOps tasks required for compliance

   While installing IBM Cloud Pak for AIOps you do need to complete some other required tasks to ensure you enable your environment to support FIPS.

3. Optional. Conditional tasks required for compliance

   Depending on the connections that you plan to create for your IBM Cloud Pak for AIOps environment, you need to complete other tasks to enable FIPS support. For instance if you are connecting IBM Cloud Pak for AIOps to IBM Tivoli Netcool/OMNIbus, you need to configure the connection to support FIPS.

Installing Red Hat OpenShift Container Platform and storage tasks required for compliance

1. Enable FIPS support for your Red Hat OpenShift Container Platform cluster
2. Enable FIPS support for your storage

1. Enable FIPS support for your Red Hat OpenShift Container Platform cluster

You must enable FIPS support on your Red Hat OpenShift Container Platform cluster before you proceed with installing IBM Cloud Pak for AIOps. When you are installing IBM Cloud Pak for AIOps, the installation process automatically detects whether FIPS support is enabled on your Red Hat OpenShift Container Platform cluster. If FIPS is enabled on Red Hat OpenShift Container Platform, FIPS support is automatically enabled for IBM Cloud Pak for AIOps.

To enable FIPS for IBM Cloud Pak for AIOps, you must first enable FIPS support on your Red Hat OpenShift Container Platform cluster by completing the following tasks as part of installing Red Hat OpenShift Container Platform:

1. Enable FIPS mode on all of your nodes. For more information, see the Red Hat OpenShift Container Platform documentation about Support for FIPS cryptography

2. Install Red Hat OpenShift Container Platform in FIPS mode. For more information, see the Red Hat OpenShift Container Platform documentation about Installing a cluster in FIPS mode.

   Restriction: FIPS is supported only on x86_64 hardware.

3. Configure TLS protection for your node-to-node communication. Node-to-node communication must be TLS protected at all times.

Recommended: Configure IPSec tunnels for communication between nodes in your cluster. IPSec is the validated and recommended method for implementing security across nodes for IBM Cloud Pak for AIOps. If you require a different method to secure your nodes, you can use your preferred method.

With IPsec enabled, all network traffic between nodes on the OVN-Kubernetes Container Network Interface (CNI) cluster network travels through an encrypted tunnel. IPsec is disabled by default when you install OpenShift 4.x clusters. IPsec encryption can be enabled only during cluster installation and cannot be disabled after it is enabled.

For more information about configuring IPsec encryption on Red Hat OpenShift Container Platform, see the Red Hat OpenShift Container Platform documentation about Configuring IPsec encryption.

2. Enable FIPS support for your storage

If your deployment's storage must be FIPS compliant, enable any FIPS settings and support for your chosen storage. Refer to your storage provider's documentation to ensure that your storage meets this requirement.

Red Hat® OpenShift® Data Foundation and FIPS

Red Hat® OpenShift® Data Foundation uses FIPS 140-2 certified cryptographic modules. You must use cluster wide encryption, and not encrypt persistent volumes individually. This is because Red Hat OpenShift Data Foundation persistent volume encryption is only available for block storage, and one of IBM Cloud Pak for AIOps's components, Zen, requires file storage. For more information, see the topic *Cluster-wide encryption* in the Red Hat OpenShift Data Foundation documentation.

Portworx and FIPS

Portworx uses FIPS 140-2 certified cryptographic modules. Portworx can encrypt each of the persistent volumes individually, or can encrypt the whole storage cluster. For more information, see the topic *Create encrypted PVCs* in the Portworx documentation.

Installing IBM Cloud Pak for AIOps tasks required for compliance

With your storage configured, continue your installation of IBM Cloud Pak for AIOps on your cluster.

Ensure that you complete the required installation tasks for compliance while you are installing IBM Cloud Pak for AIOps. These tasks must be completed while you are initially installing IBM Cloud Pak for AIOps.

For more information about, and instructions, for installing IBM Cloud Pak for AIOps, see:

- For more information about installing on specific cloud platforms, see Supported cloud environments for installing IBM Cloud Pak for AIOps.
- For more information about installing IBM Cloud Pak for AIOps overall, see Production installation. If you are installing a starter installation, see Starter installation.

Important: You must ensure that you have run the *Verify the installation* step for your chosen installation method, and that your deployment is healthy before you proceed.

Enable two-factor authentication (2FA)

To secure your environment with multi-factor authentication, you can configure single sign-on (SSO) between IBM Cloud Pak for AIOps and an identity provider that supports 2FA, such as IBM Security Verify. The identity provider's 2FA mechanism can then be used to authenticate IBM Cloud Pak for AIOps user login. Consult your chosen identity provider's documentation to setup 2FA.

For more information, see Configuring single sign-on.

For more information about IBM Security Verify, see IBM Security Verify.

Blog: A blog that provides an example of setting up single sign-on between a IBM Cloud Pak and IBM Security Verify is here: Tutorial: IBM Cloud Pak single sign-on (SSO) integration with IBM Security Verify.

Conditional tasks required for compliance

The following tasks are required depending on the connections you plan to create for your IBM Cloud Pak for AIOps environment:

- IBM Cloud Pak for AIOps and Netcool connection compliance requirement: Enabling FIPS support

IBM Cloud Pak for AIOps and Netcool connection compliance requirement: Enabling FIPS support

If you plan to connect IBM Cloud Pak for AIOps with IBM Tivoli Netcool/OMNIbus, you need to configure your connection to enable FIPS support. For more information, see Enabling Federal Information Processing Standards (FIPS) support for a Netcool connection.

Enabling Federal Information Processing Standards (FIPS) support for Infrastructure Automation

Learn how to enable FIPS and run Infrastructure Automation on a FIPS-compliant system.

To enable support you need to install Infrastructure Automation on a Red Hat OpenShift Container Platform cluster that is running in a FIPS enabled mode. You also need to complete some tasks before you begin installing Infrastructure Automation on your cluster and complete other tasks while you are installing Infrastructure Automation. Depending on whether you plan to enable or set up some optional connections, you might need to further complete tasks after your finish installing to ensure your connections support FIPS.

Important: You can only enable FIPS support when you are installing. You cannot upgrade a non-FIPS enabled environment to a FIPS enabled environment. You also cannot backup an Infrastructure Automation from a non-FIPS enabled environment and restore it into a FIPS enabled environment.

To enable FIPS support, complete the following tasks during your overall cluster and Infrastructure Automation installation:

1. Installing Red Hat OpenShift Container Platform and storage tasks required for compliance.

   These tasks must be completed when you are installing your Red Hat OpenShift Container Platform cluster and configuring your storage before you begin to install Infrastructure Automation on your OpenShift cluster.

2. Installing Infrastructure Automation tasks required for compliance

   While installing Infrastructure Automation you do need to complete some other required tasks to ensure you enable your environment to support FIPS:

3. Optional. Conditional tasks required for compliance

   Depending on the components that you plan to install in, or integrate with, your Infrastructure Automation environment, you need to complete other tasks to enable FIPS support. For instance if you plan to enable Managed services, you need to configure the deployment to support FIPS.

Installing Red Hat OpenShift Container Platform and storage tasks required for compliance

1. Enable FIPS support for your Red Hat OpenShift Container Platform cluster
2. Enable FIPS support for your storage

1. Enable FIPS support for your Red Hat OpenShift Container Platform cluster

You must enable FIPS support on your Red Hat OpenShift Container Platform cluster before you proceed with installing Infrastructure Automation. When you are installing Infrastructure Automation, the installation process automatically detects whether FIPS support is enabled on your Red Hat OpenShift Container Platform cluster. If FIPS is enabled on Red Hat OpenShift Container Platform, FIPS support is automatically enabled for Infrastructure Automation.

To enable FIPS for Infrastructure Automation, you must first enable FIPS support on your Red Hat OpenShift Container Platform cluster by completing the following tasks as part of installing Red Hat OpenShift Container Platform:

1. Enable FIPS mode on all of your nodes. For more information, see the Red Hat OpenShift Container Platform documentation about Support for FIPS cryptography

2. Install Red Hat OpenShift Container Platform in FIPS mode. For more information, see the Red Hat OpenShift Container Platform documentation about Installing a cluster in FIPS mode.

   Restriction: FIPS is supported only on x86_64 hardware.

3. Configure TLS protection for your node-to-node communication. Node-to-node communication must be TLS protected at all times.

   Recommended: Configure IPSec tunnels for communication between nodes in your cluster. IPSec is the validated and recommended method for implementing security across nodes for IBM Cloud Pak for AIOps. If you require a different method to secure your nodes, you can use your preferred method.

   With IPsec enabled, all network traffic between nodes on the OVN-Kubernetes Container Network Interface (CNI) cluster network travels through an encrypted tunnel. IPsec is disabled by default when you install OpenShift 4.x clusters. IPsec encryption can be enabled only during cluster installation and cannot be disabled after it is enabled.

   For more information about configuring IPsec encryption on Red Hat OpenShift Container Platform, see the Red Hat OpenShift Container Platform documentation about Configuring IPsec encryption.

2. Enable FIPS support for your storage

If your deployment's storage must be FIPS compliant, enable any FIPS settings and support for your chosen storage. Refer to your storage provider's documentation to ensure that your storage meets this requirement.

Red Hat® OpenShift® Data Foundation and FIPS

Red Hat® OpenShift® Data Foundation uses FIPS 140-2 certified cryptographic modules. You must use cluster wide encryption, and not encrypt persistent volumes individually. This is because Red Hat OpenShift Data Foundation persistent volume encryption is only available for block storage, and one of IBM Cloud Pak for AIOps's components, Zen, requires file storage. For more information, see the topic *Cluster-wide encryption* in the Red Hat OpenShift Data Foundation documentation.

Portworx and FIPS

Portworx uses FIPS 140-2 certified cryptographic modules. Portworx can encrypt each of the persistent volumes individually, or can encrypt the whole storage cluster. For more information, see the topic *Create encrypted PVCs* in the Portworx documentation.

Installing Infrastructure Automation tasks required for compliance

With your storage configured, continue your installation of Infrastructure Automation on your cluster.

Ensure that you complete the required installation tasks for compliance while you are installing Infrastructure Automation. These tasks must be completed while you are initially installing Infrastructure Automation.

For more information about, and instructions, for installing Infrastructure Automation overall, see Installating Infrastructure Automation.

Enable two-factor authentication (2FA)

To secure your environment with multi-factor authentication, you can configure single sign-on (SSO) between Infrastructure Automation and an identity provider that supports 2FA, such as IBM Security Verify. The identity provider's 2FA mechanism can then be used to authenticate Infrastructure Automation user login. Consult your chosen identity provider's documentation to setup 2FA.

For more information, see Configuring single sign-on.

For more information about IBM Security Verify, see IBM Security Verify.

Blog: A blog that provides an example of setting up single sign-on between a IBM Cloud Pak and IBM Security Verify is here: Tutorial: IBM Cloud Pak single sign-on (SSO) integration with IBM Security Verify.

Conditional tasks required for compliance

The following tasks are required depending on the components that you plan to enable for your Infrastructure Automation environment:

- Deploy Infrastructure Automation - Managed services in FIPS mode

Deploy Infrastructure Automation - Managed services in FIPS mode

If you need to deploy Managed services in your FIPS enabled environment, you can deploy the component after you complete your install by editing the **IAConfig** settings. This task can be completed after you complete your initial installation, but needs to be completed when you are initially deploying Managed services.

1. Use the OpenShift (**oc**) CLI to log in to your OpenShift cluster and switch to the project (namespace) where Infrastructure Automation is installed.

2. Run the following command to open the **IAConfig** configuration for editing:

   ```
   oc -n <namespace> edit IAConfig
   ```

   Where **<namespace** is the project (namespace) where Infrastructure Automation is installed.

3. Search for the **spec** section and edit the configuration to include the settings to deploy Managed services (**ibm-management-cam-install**) in FIPS mode (**enableFIPS: true**). Your updated configuration for deploying managed services can resemble the following example spec:

   ```
   spec:
       imagePullSecret: ibm-entitlement-key
       infraAutoComposableComponents:
       - enabled: true
         name: ibm-management-im-install
         spec: {}
       - enabled: true
         name: ibm-management-cam-install
         spec:
           manageservice:
             global:
               enableFIPS: true
             license:
               accept: true
       license:
         accept: true
   ```

Enabling Federal Information Processing Standards (FIPS) support for an IBM Tivoli Netcool/OMNIbus connection

If you plan to connect IBM Cloud Pak for AIOps with IBM Tivoli Netcool/OMNIbus with the Netcool connector, you can configure your connection to enable support for Federal Information Processing Standards (FIPS).

Prerequisites

- You must have an existing FIPS-compliant installation of IBM Cloud Pak for AIOps. You cannot enable FIPS support for the connector on a noncompliant system.

  For more information about installing IBM Cloud Pak for AIOps to be FIPS-compliant, see Enabling support for Federal Information Processing Standards (FIPS).

- You must have an existing FIPS-compliant installation of IBM Tivoli Netcool/OMNIbus.

Procedure

To enable FIPS mode in IBM Tivoli Netcool/OMNIbus, complete the following steps on the host where IBM Tivoli Netcool/OMNIbus (on premises ObjectServer) is running:

1. Stop the ObjectServer pair and the IBM Tivoli Netcool/OMNIbus bidirectional gateway.

   - To stop the ObjectServer, complete the following steps:

     - Open the ObjectServer SQL prompt:

       ```
       $OMNIHOME/bin/nco_sql -server <OS_NAME> -user root -password <plain-text-pwd>
       ```

     - On the SQL prompt, run the following command:

       ```
       > alter system shutdown;
       > go
       ```

     Note: After you enter **go**, the SQL prompt disconnects as the ObjectServer is exiting.

   - To stop the bidirectional gateway, run the following command:

     ```
     kill -15 <pid>
     ```

     where **pid** is Process ID.

     Note: Use the **ps -ef | grep nco_g_objserv_bi** command to get the gateway's pid.

2. Enter the following information in the properties file of both the ObjectServers:

   ```
   PasswordEncryption   : 'AES'
   ConfigKeyFile        : '<key_file>'
   ```

   Note: To generate the key file, run the following command on the host:

   ```
   $OMNIHOME/bin/nco_keygen -o <key_file> -l 256
   ```

3. Start the ObjectServer pair and update the same FIPS-compliant password for the root user in each ObjectServer.

   1. To start the ObjectServer, run the following command:

      ```
      $OMNIHOME/bin/nco_objserv -name <OS_NAME> -propsfile <objectserver-properties-file>
      &
      ```

   2. To open the ObjectServer SQL prompt, run the following command:

      ```
      $OMNIHOME/bin/nco_sql -server <OS_NAME> -user root -password <plain-text-pwd>
      ```

   3. To alter root password, run the following SQL command:

      ```
      > alter user 'root' set password '<plain_text_pwd>';
      > go
      ```

   4. To verify that the Passwd column contains AES-encrypted passwords with 24 characters, run the following SQL command:

      ```
      > select UserName,Passwd from security.users;
      > go
      ```

4. Generate the encrypted IBM Tivoli Netcool/OMNIbus root password by using the ConfigKeyFile.

   - To create the encrypted password, run the following command:

     ```
     $OMNIHOME/bin/nco_aes_crypt -c AES_FIPS -k string -o <encrypted_file> -k <key_file>
     <root_plain_text_pwd>
     ```

     Note: For more information about generating a key in a key file, see Generating a key in a key file.

5. Apply the configurations in the properties file of the bidirectional gateway:

   ```
   Gate.ObjectServerA.Password  : '<encrypted_root_pwd>'
   Gate.ObjectServerB.Password  : '<encrypted_root_pwd>'
   ConfigKeyFile                : '<key_file>'
   ```

6. Place the following files in the **$NCHOME/etc/security** directory to enable FIPS mode in IBM Tivoli Netcool/OMNIbus:

   - **fips.conf**:

     ```
     SP800_131MODE=TRUE
     TLS12_ONLY=TRUE
     ```

   - **sslciphers.conf**:

     ```
     sslv3_remove=_
     tls10_remove=_
     tls11_remove=_
     tls12_remove="TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH_AES_128_CBC_SHA,TLS_RSA_WITH
     _AES_256_CBC_SHA,TLS_RSA_WITH_AES_128_GCM_SHA256,TLS_RSA_WITH_AES_256_GCM_SHA384,TLS
     _RSA_WITH_AES_128_CBC_SHA256,TLS_RSA_WITH_AES_256_CBC_SHA256"
     ```

7. Start the ObjectServer pair and the bidirectional gateway.

   - To start the ObjectServer, run the following command on the host:

     ```
     $OMNIHOME/bin/nco_objserv -name <OS_NAME> -propsfile <objectserver-properties-file>
     &
     ```

     For example,

     ```
     $OMNIHOME/bin/nco_objserv -name AGG_P -propsfile $OMNIHOME/etc/AGG_P.props &
     ```

   - To start the bidirectional gateway, run the following command on the host:

     ```
     $OMNIHOME/bin/nco_g_objserv_bi -propsfile <gateway-properties-file> &
     ```

     For example,

     ```
     $OMNIHOME/bin/nco_g_objserv_bi -propsfile $OMNIHOME/etc/AGG_GATE.props &
     ```

Audit logging

The audit logging capabilities within IBM Cloud Pak for AIOps provide you with the capability to track changes that were made to your system, when the changes were made, and by whom. The recorded audit logs include the record of actions that were completed or attempted by users or services of your IBM Cloud Pak for AIOps system. This auditing helps you with ensuring accountability, trackability, and regulatory compliance for data access, modification, and security within your IBM Cloud Pak for AIOps environment.

If you require that your environment supports the Federal Regulatory Compliance Standard (FISMA), you need to ensure that your environment has audit logging.

What is audited in IBM Cloud Pak for AIOps?

IBM Cloud Pak for AIOps capabilities audit the following types of actions, which can return, manage, manipulate, or access sensitive data:

- Successful and unsuccessful login attempts
- Authentication checks
- Authorization checks
- Account management events
- Object access
- Policy changes
- Privileged functions
- Data access
- Data changes
- Process tracking
- System events

- Administrator activities
- Permission changes

For more information and details about what is logged for IBM Cloud Pak for AIOps, see Logged actions and data for auditing.

How do you configure audit logging?

For IBM Cloud Pak for AIOps capabilities, audit logging is configured and enabled by default.

For Infrastructure Automation, you need to enable audit logging during installation. For the list of audit-related installation parameters, see Installation parameters.

For Red Hat OpenShift Container Platform audit logging, see the Red Hat OpenShift Container Platform documentation Viewing audit logs.

How do you view a record of audit logs?

- For IBM Cloud Pak for AIOps capabilities, the audit logs for all recorded actions are written to standard output **stdout** in JSON format. You can view an audit log for a pod by using the **oc logs** command.

  If you want to, or need to, view multiple audit logs, you can ingest and view the logs with a log aggregator, such as the logging subsystem for Red Hat OpenShift. With this logging subsystem, you can aggregate all the logs from your OpenShift Container Platform cluster, including IBM Cloud Pak for AIOps audit logs. You can then view the logs with the Kibana web console. For more information about the logging subsystem for Red Hat OpenShift, see the Red Hat OpenShift documentation Installing the logging subsystem for Red Hat OpenShift.

  You can also use other services for log aggregation, such as Falcon Logscale or an Elasticsearch, Logstash, and Kibana (ELK) stack.

  The following procedure shows how to use the **oc** command to view API audit logs, which are stored in the Kong pod logs:

  1. Run the following command to get the Kong pod name:

     ```
     kongpod=$(oc get pod | grep "gateway-kong-" | egrep  -v "config-svc|post-install" | awk '{print $1}')
     ```

  2. Then, run the following command to view the audit log in the **proxy** container of the Kong pod:

     ```
     oc logs $kongpod -c proxy | grep eventTime
     ```

- For Infrastructure Automation, Infrastructure management appliances use the systemd logger, journalctl for audit logging. Appliances that run UI workers log the UI requests. Appliances that run the web services role log the API requests.

  Logging for podified deployments use the standard output **stdout** logger. You can view these logs through an **oc logs** command. To view a log, you need to specify the correct UI or web service pod.

What is included in an audit log

The audit log messages for a recorded action or event in IBM Cloud Pak for AIOps include information to identify the action or event that occurred, when the action or event occurred, the action or event source and outcome, and the identity of the associated user or service.

For the more information about what is included in audit log messages, including example messages, see Audit log messages.

How do you forward audit logs?

You can use the IBM Cloud Pak foundational services Fluentd to forward audit logs.

Fluentd is a log collector that is available through IBM Cloud Pak foundational services. This collector uses input and output plug-ins to collect data from multiple sources and to distribute or send data to multiple destinations.

Fluentd can be configured to send logs to an enterprise SIEM tool such as IBM QRadar. For more information, see Audit logging version 3.7.0.

Logged actions and data for auditing

For audit logging, IBM Cloud Pak for AIOps capabilities log information about numerous user interactions, system processes, and events, which can each return, manage, manipulate, or access sensitive data. The following lists, detail the actions and data that is recorded in audit logs.

- IBM Cloud Pak for AIOps
- Infrastructure Automation

IBM Cloud Pak for AIOps

IBM Cloud Pak for AIOps capabilities audit the following types of actions, which can return, manage, manipulate, or access sensitive data:

- Successful and unsuccessful login attempts
- Authentication checks These checks run when a user attempts to log in to the UI. The record for these checks includes whether the attempt was successful. These checks do not include checks for internal system service-to-service authentication actions, such as when a secret is used to authenticate to a database.
- Authorization checks These checks run to determine whether a user has the required role and permission for completing an action.
- Account management events These events include actions for adding, changing, removing users, and managing credentials and role-based access control.
- Object access This access includes the access of data objects through the IBM Cloud Pak for AIOps UI or APIs.
- Policy changes The auditing of these changes includes policy changes for the management of user access controls, such as access policies. This auditing does not include event suppression policies.
- Privileged functions This auditing records changes for administrator access controls, such as access policies and their ability to run administrator-level tasks.
- Data access This auditing records the read-access any of data, such as viewing data on a UI page. The viewing of the following data through the UI or API is audited:
    - Connections
        - Viewing the set of configured connections
        - Viewing the status of a connection
        - Viewing the configuration of a connection
    - AIOps insights
        - Viewing aggregate usage information, such as information about the number of resources, events, incidents, runbooks
    - AI Models
        - Viewing the set of configured AI model definitions
        - Viewing a AI model definition's status
        - Viewing a AI model definition's configuration
        - Viewing the status and details of an AI model version
    - Resources
        - Viewing the list of resources, resources groups, or applications
        - Searching resources, resource groups, or applications
        - Viewing the configuration of a resource, resource group, or application
        - Viewing the topology of an application
    - Incidents and alerts
        - Viewing the list of incidents or alerts
        - Viewing incident or alert details

    - Automations
        - Viewing the list of policies
        - Viewing a specific policy
        - Viewing the list of runbooks
        - Viewing a specific runbook
        - Viewing the list of automation actions
        - Viewing a specific automation action
- Data changes This auditing records actions for creating or changing data. Changes to the following types of data are recorded:

    - Connections
        - Creating or editing a connection
    - AI Models
        - Creating or editing an AI model definition
        - Creating an AI model version through AI training
        - Editing the state of an AI model version
    - Resources
        - Creating or editing a resource, resource group, or application
    - Alerts & Incident
        - Changing the state of an alert, including by policies
        - Editing an alert journal
        - Changing the incident details, such as the incident owner or state
    - Automations
        - Creating or editing a policy, runbook, or action
- Data deletions This auditing records the deletion of the following types of data:

    - Connections
        - Deleting a connection
    - AI models
        - Deleting an AI model definition or AI model version
    - Resources
        - Deleting a resource, resource group, or application
    - Automations
        - Deleting a policy, runbook, or action
- Process tracking This auditing records the action that caused a policy to run. A policy action is not caused by a user interaction. Instead, the action is caused by the policy service, with the data object for which the policy is applied being the target. Audit logs record an entry (policy ID, ID of the entity the policy ran on) whenever a policy runs due to one of the following actions occurring:

    - aiops.ibm.com/trigger/event-received
    - aiops.ibm.com/trigger/alert-pre-create
    - aiops.ibm.com/trigger/alert-created
    - aiops.ibm.com/trigger/alert-pre-update
    - aiops.ibm.com/trigger/alert-pre-delete
    - aiops.ibm.com/trigger/incident-pre-create
    - aiops.ibm.com/trigger/incident-created
    - aiops.ibm.com/trigger/incident-pre-update
    - aiops.ibm.com/trigger/incident-pre-delete

  One or more of these actions can occur when one of the following processes run:

    - Alert suppression through a policy
    - Alert creation through a policy
    - Correlating an alert to another alert through a policy
    - Creating an incident through a policy
    - Assigning or running an action through the UI, API, or policy
    - Assigning or running a runbook through the UI, API, or policy
    - Running an Ansible, HTTP or SSH automation action on an external system
    - Creating or updating an incident ticket through a policy
    - Creating or updating a ChatOps message through a policy
    - Determining a change risk score through a policy
    - Updating a change ticket with a change risk score through a policy

    - Providing model feedback
- System events This auditing records system and maintenance tasks that are not included within the preceding categories, such as the following actions:
    - Back up and restore
- Administrator activities This auditing records activities that require a user with an administrator role, including the following activities:
    - Connections
        - Viewing the set of configured connections
        - Viewing the status of a connection
        - Viewing the configuration of a connection
        - Creating, editing, or deleting a connection
    - AI models
        - Viewing the set of configured AI model definitions
        - Viewing a AI model definition's status
        - Viewing a AI model definition's configuration
        - Viewing the status and details of an AI model version
        - Creating, editing, or deleting an AI model definition
        - Creating, editing, or deleting an AI model version
    - Resources
        - Viewing the configuration of a resource, resource group, or application
        - Creating, editing, or deleting a resource, resources group, or application
    - Automations
        - Viewing a specific runbook
        - Viewing the list of automation actions
        - Viewing a specific automation action
        - Creating, editing, or deleting a policy, runbook, or action
- Permission changes Records of the changes and management of user permissions.

Infrastructure Automation

Infrastructure Automation capabilities audit the following types of actions and data:

- Infrastructure management
- Managed services

Infrastructure management

Infrastructure management audits UI and API interactions. This logging includes UI privilege checks, invalid session requests through a session timeout or reloading a page after a user logs out.

Managed services

Managed services logging includes records for the following actions:

- Logging in or out of the UI
- Creating, editing, or deleting a cloud connection
- Creating, editing or deleting a DataTypes in shared parameters
- Creating, editing or deleting a DataObjects in shared parameters
- Adding or editing namespace for a DataObjects in shared parameters
- Creating, editing or deleting dynamic parameters
- Deploying a stack
- Modifying an existing stack (plan and apply)
- Modifying an existing stack record attributes
- Destroying or deleting a stack
- Creating, deleting an IAAS resource task
- Creating, editing, or deleting an IAAS resource settings
- Creating, editing, or deleting a template
- Creating, editing, or deleting a template version
- Creating, update or deleting provider engine
- Creating, update or deleting providers plugins

- Creating, editing, duplicating, retiring or deleting a service
- Adding or editing namespaces for a service
- Creating or deleting a service plan
- Creating, editing, or deleting a service category
- Adding or editing a service version
- Publishing, retiring, unretiring, duplicating, or deleting a service version
- Pushing to, or Pulling from, a Git service version
- Deploying a service-instance
- Resuming or Retrying a service-instance
- Terminating or deleting a service-instance
- Invoking or terminating a bind action for a service-instance
- Invoking a custom action for a service-instance
- Generating update plan for a service-instance
- Updating a service-instance
- Getting Upgrade or Downgrade service versions for a service-instance
- Generating service version change plan for a service-instance
- Running service version change for a service-instance
- Creating, editing, or deleting an independent custom action category
- Creating, editing, deleting, or invoking independent custom action
- Set or Unset namespace for independent custom action
- Lock or Unlock an independent custom action
- Creating, editing, or deleting an SMTP connection
- Adding or Deleting subscription store configuration

Audit log messages

For IBM Cloud Pak for AIOps capabilities, the log message entries for a recorded action or event include information to identify the action or event that occurred, when the action or event occurred, the action or event source and outcome, and the identity of the associated user or service.

Note: Some audit-specific fields, such as the username, are part of a generic message field and are not broken out into a dedicated field.

For Infrastructure Automation Infratructure management, the audit log messages include the HTTP method, username, role, request ID, session ID, requested URL path, and details for any specific privilege checks.

For example, a request to load the dashboard at **`/dashboard/show`** logs the **`show`** action and the privilege check (**`dashboard_view`**) for that page. Some pages can have multiple privilege checks as the UI determines what data a user can view or act upon.

Log messages

Log messages can include the following details:

Table. Log message details

Field Name	Required (IBM Cloud Pak for AIOps)	Required (UI)	Required (API)	Description	Example	Jsonpath in kong logging (API)

Field Name	Required (IBM Cloud Pak for AIOps)	Required (UI)	Required (API)	Description	Example	Jsonpath in kong logging (API)
logType	Yes	Yes	Yes	An indicator of the fact that this is an audit message. This is used for allowing easy identifying audit versus other log messages. This is always set to `audit`	`audit`	
id	No	No	No	The unique ID for the request	`271bfd04-a625-48ec-b33c-c7e9958f646c`	
initiator.id	Yes	Yes	Yes	ID of the identity or user that initiated the action.	User IAM ID API key	`.request.headers.audit-initiator-id`
initiator.name	Yes	Yes	Yes	Name of the identity or user that initiated the action.	username for users `system` `policy-service` for policy actions.	`.request.headers.audit-initiator-name`
initiator.typeURI	Yes	Yes	Yes	CADF URI description of the initiator type. For UI audit logs, this URI is most likely the user.	`security/account/user` for cookie or user initiated actions `security/account/service` or `security/account/admin` for apikey or service or system initiated actions.	`.request.headers.audit-initiator-typeuri`
initiator.credential.type	Yes	Yes	Yes	Type of authentication provided by the initiator. For UI audit logs, the value is often `cookie`.	`token` `user` `apikey` `cookie` `unknown`	`.request.headers.audit-initiator-credential-type`
initiator.credential.identity_status	No	No	No	Status of the identity authentication for the initiator or user.	`Confirmed` `Denied` `Unknown`	`request.headers.audit-initiator-credential-identity_status`
initiator.host.address	No	No	No	Source IP from where the request originated.	IP address of the Red Hat OpenShift Load Balancer.	`.client_ip`
initiator.host.agent	N/A	N/A	Yes	The client type of API request	`curl/7.61.1` `HTTPie/2.4.0`	`.request.headers.user-agent`

Field Name	Required (IBM Cloud Pak for AIOps)	Required (UI)	Required (API)	Description	Example	Jsonpath in kong logging (API)
target.name	Yes	Yes	Yes	The name of the service that is accepting the request. For UI audit logs, this is populated as an environment variable in the deployed UI server (`AUDIT_SERVICE_NAME`)	`service; resource api.acme.com`	`.service.host`
target.id	Yes	No	Yes	CRN of the service accepting this request. For UI audit logs, this value is populated as an environment variable in the deployed UI server (`AUDIT_SERVICE_CRN`)	`b7dd7db9-45f2-4c05-abb5-01466c2abcdf`	`.service.host`
target.typeURI	Yes	No	Yes	This is a constant value that is stored in `AUDIT_RESOURCE_CADF_URI`.	`service` for HTTP/Ansible `system` for Script	`service`
requestData.path	Yes	Yes	Yes	The request path, such as the API, which indicates the specific request being made of the service.	`api/v2/inventories/d220b490-7c75-11ed-b6f3-23388646a3b9`	`.request.uri`
requestData.type	No	Yes	No	The request method, which indicates whether this is a view, create, update, or delete.	`GET` `PUT` `DELETE` `POST`	`.request.method`
requestData.actionId	No	No	No	Automation/Action ID.	`c6ccc725-3e54-493d-a114-de5130e7c64b`	
action	Yes	No	Yes	The action that triggers an event. This is superceded by `requestData.method`.	`create` `update` `delete` `deploy` `authenticate`	`view` `create`

Field Name	Required (IBM Cloud Pak for AIOps)	Required (UI)	Required (API)	Description	Example	Jsonpath in kong logging (API)
outcome	Yes	Yes	Yes	Was the action succesful?	`success` `pending` `failure` `unknown`	`success`
reason.reasonCode	Yes	Yes	Yes	The HTTP response code of the result.	`200` for success.	`.response.status`
reason.reasonSize	Yes	Yes	Yes	The HTTP response size of the result.	`116`	`.response.size`
reason.message	Yes	No	Yes	The HTTP response message explaining the result. For API, Follow Format 'action [custom data] [outcome]'	`OK` for success.	`view success`
eventTime	Yes	Yes	Yes	The time, date, and time zone of the event in Coordinated Universal Time (UTC)	`2022-11-14T11:28:54.753Z`	`.started_at`
attachments.content.correlation_id	No	No	No	An optional ID that can be used to correlate multi-part audit logs.	`ccccccc-eeee-dddd-aaaa-bbbbbbbbbbbb`	

Examples

Example: Kong logging payload (JSON format)


```
{
    "latencies": {
        "request": 22,
        "kong": 3,
        "proxy": 19
    },
    "service": {
        "host": "aiops-topology-rest-observer.katamari.9104.svc",
        "created_at": 1669258991,
        "connect_timeout": 60000,
        "id": "3bf5089f-1aff-52e9-b41f-21f1e357b072",
        "protocol": "https",
        "read_timeout": 60000,
        "port": 80,
        "path": "/",
        "updated_at": 1669258991,
        "write_timeout": 60000,
    "name": "katamari.aiops-topology-rest-observer.pnum-9104",
    "enabled": true,
        "retries": 5,
        "ws_id": "0dc6f45b-8f8d-40d2-a504-473544ee190b"
    },
    "request": {
        "querystring": {},
        "size": 5682,
        "uri": "/aiops/api/application-manager/topology-rest-observer/v1/healthcheck",
```

```
        "url": "https://cpd-katamari.apps.cp4waiops-svl-
04063015.mysite.ibm.com:443/aiops/api/application-manager/topology-rest-
observer/v1/healthcheck",
        "headers": {
            "x-forwarded-proto": "https",
            "x-consumer-id": "1181be16-0798-5424-9578-c89e63707139",
            "x-consumer-username": "api-user",
            "x-credential-identifier": "KNOXSSO",
            "accept": "application/json",
            "connection": "upgrade",
            "audit-origination": "api",
            "audit-initiator-id": "1000330999",
            "audit-initiator-name": "admin",
            "user-agent": "curl/7.61.1",
            "audit-initiator-credential-type": "token",
            "audit-initiator-credential-identity_status": "Confirmed",
            "audit-initiator-host-address": "10.254.12.2",
            "host": "cpd-katamari.apps.cp4waiops-svl-04063015.mysite.ibm.com",
            "audit-requestdata-type": "GET",
            "jwt-auth-user": "admin",
            "authorization": "REDACTED",
            "audit-requestdata-path": "/aiops/api/application-manager/topology-rest-
observer/v1/healthcheck",
            "audit-initiator-typeuri": "security/account/user"
        },
        "tls": {
            "cipher": "ECDHE-ECDSA-AES256-GCM-SHA384",
            "client_verify": "NONE",
            "version": "TLSv1.2"
        },
        "method": "GET"
    },
    "tries": [
        {
            "balancer_latency": 0,
            "port": 9104,
            "balancer_start": 1669259021798,
            "ip": "10.254.24.55"
        }
    ],
    "client_ip": "10.254.24.38",
    "upstream_uri": "/1.0/rest-observer/healthcheck",
    "started_at": 1669259021796,
    "response": {
        "headers": {
            "content-type": "application/json",
            "date": "Thu, 24 Nov 2022 03:03:41 GMT",
            "connection": "close",
            "content-length": "114",
            "x-kong-upstream-latency": "19",
            "vary": "Accept-Encoding",
            "via": "kong/3.0.0",
            "_href": "https://cpd-katamari.apps.cp4waiops-svl-
04063015.mysite.ibm.com/1.0/rest-observer/healthcheck",
            "x-kong-proxy-latency": "2"
        },
        "status": 200,
        "size": 444
    },
    "consumer": {
        "username": "api-user",
        "id": "1181be16-0798-5424-9578-c89e63707139",
        "created_at": 1669258991
    },
    "route": {
        "preserve_host": true,
        "id": "e2d7a651-47dd-5d67-bc52-77367ec2f63d",
        "paths": [
            "~/aiops/api/application-manager/topology-rest-observer/v1/(.*)"
        ],
        "created_at": 1669258991,
        "updated_at": 1669258991,
```

```
        "regex_priority": 100,
        "service": {
            "id": "3bf5089f-1aff-52e9-b41f-21f1e357b072"
        },
        "path_handling": "v0",
        "name": "katamari.topology-rest-observer-api.00",
        "ws_id": "0dc6f45b-8f8d-40d2-a504-473544ee190b",
        "strip_path": false,
        "response_buffering": true,
        "request_buffering": true,
        "protocols": [
            "http",
            "https"
        ],
        "https_redirect_status_code": 426
    },

    "authenticated_entity": {
        "id": "6a1c706c-2e2a-58e2-9db8-4740b7959fa3"
    }
}
```


Transformed audit logging record sample:


```
{
    "outcome": "success",
    "eventTime": "2023-02-03T06:13:17Z",
    "reason": {
        "reasonSize": 438,
        "message": "view success",
        "reasonCode": 200
    },
    "initiator": {
        "name": "admin",
        "host": {
            "agent": "curl/7.61.1",
            "address": "10.254.36.42"
        },
        "id": "1000330999",
        "typeURI": "security/account/user",
        "credential": {
            "idenity_status": "Confirmed",
            "type": "token"
        }
    },
    "requestData": {
        "type": "GET",
        "path": "/aiops/api/application-manager/topology-rest-observer/v1/healthcheck"
    },
    "target": {
        "id": "aiops-topology-rest-observer.katamari.9104.svc",
        "typeURI": "service",
        "name": "aiops-topology-rest-observer.katamari.9104.svc"
    },
    "action": "view"
}
```


Example: Policy action audit log

The following message is an example audit log message for policy actions.


```
{
  "mdc": {
    "action": "execute",
    "eventTime": "2023-03-07T18:11:48.948822949Z",
    "id": "3451a11e-45fc-4b00-8a8f-912a88b156d3",
    "initiator.credential.type": "unknown",
    "initiator.id": "crn:v1:cp4waiops:aiops:::policy:policy-service",
    "initiator.name": "policy-service",
    "initiator.typeURI": "security/account/service",
    "key": "{\"ce-partitionkey\":\"b6c78d1e-9d71-403f-aab2-e8db9cc30202\"}b345a777-3dbc-49ed-
90a0-97cde7b9fcbc",
```

```
    "logType": "audit",
    "outcome": "success",
    "policyId": "84722b70-bce9-11ed-bec2-81518fc878b2",
    "policyRequestId": "3451a11e-45fc-4b00-8a8f-912a88b156d3",
    "reason.message": "Received policy execution result for key: {\"ce-
partitionkey\":\"b6c78d1e-9d71-403f-aab2-e8db9cc30202\"}b345a777-3dbc-49ed-90a0-97cde7b9fcbc,
status: COMPLETE",
    "reason.reasonCode": "200",
    "requestData.path": "84722b70-bce9-11ed-bec2-81518fc878b2",
    "target.id": "{\"ce-partitionkey\":\"b6c78d1e-9d71-403f-aab2-e8db9cc30202\"}b345a777-3dbc-
49ed-90a0-97cde7b9fcbc",
    "target.typeURI": "aiops.ibm.com/trigger/event-received"
  },
  "@version": 1,
  "source_host": "aiops-ir-lifecycle-eventprocessor-ep-taskmanager-0",
  "message": "Audit policy execution result",
  "thread_name": "policy-exec::schedule-policy-execution (1/2)#0",
  "@timestamp": "2023-03-07T18:11:48.948+0000",
  "level": "INFO",
  "logger_name": "com.ibm.aiops.lifecycle.policy.scheduling.PolicySchedulerProcessFunction"
}
```


Example: Infrastructure Management audit log

The following example log shows an audit log that includes log entries for successful requests, logon failures, requests that failed due to missing product features, and an invalid (stale) session.


```
[----] W, [2023-01-27T10:02:29.500256 #17089:5a398]  WARN -- audit: <AuditFailure> Username
[blah], from: [Base.audit_failure], Authentication failed for userid blah
[----] I, [2023-01-27T10:02:36.633339 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username
[admin], from: [Base.audit_success], User admin successfully validated by EVM
[----] I, [2023-01-27T10:02:36.636510 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username
[admin], from: [Base.audit_success], Authentication successful for user admin
[----] I, [2023-01-27T10:02:37.045266 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [e35c5068-9cee-41c9-89c7-a12024b61e82],
Method [GET], Path [/dashboard/show] Action: show
[----] I, [2023-01-27T10:02:37.045501 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [e35c5068-9cee-41c9-89c7-a12024b61e82],
Method [GET], Path [/dashboard/show] Features checked: dashboard_view
[----] I, [2023-01-27T10:02:39.294107 #17089:66d00]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [387576de-5e70-4517-867e-9d10259b8b73],
Method [GET], Path [/dashboard/widget_chart_data/48] Action: widget_chart_data
[----] I, [2023-01-27T10:02:39.294856 #17089:66d00]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [387576de-5e70-4517-867e-9d10259b8b73],
Method [GET], Path [/dashboard/widget_chart_data/48] Features checked: dashboard_view
[----] I, [2023-01-27T10:02:39.440698 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [4f302068-b92a-497f-8c20-4c3e99c33057],
Method [GET], Path [/dashboard/widget_report_data/19] Action: widget_report_data
[----] I, [2023-01-27T10:02:39.443928 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [4f302068-b92a-497f-8c20-4c3e99c33057],
Method [GET], Path [/dashboard/widget_report_data/19] Features checked: dashboard_view
[----] I, [2023-01-27T10:02:39.647895 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [efaa6a1e-ad5c-4f9f-9daf-e95be6cd1af5],
Method [GET], Path [/dashboard/widget_report_data/17] Action: widget_report_data
[----] I, [2023-01-27T10:02:39.649264 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [efaa6a1e-ad5c-4f9f-9daf-e95be6cd1af5],
Method [GET], Path [/dashboard/widget_report_data/17] Features checked: dashboard_view
[----] I, [2023-01-27T10:02:39.934904 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [5912f12d-3a7c-497e-9766-9c62e18ea247],
Method [GET], Path [/dashboard/widget_menu_data/34] Action: widget_menu_data
[----] I, [2023-01-27T10:02:39.935245 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [5912f12d-3a7c-497e-9766-9c62e18ea247],
Method [GET], Path [/dashboard/widget_menu_data/34] Features checked: dashboard_view
[----] I, [2023-01-27T10:02:39.990627 #17089:66d00]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [1a7b7ec1-8d37-44ee-bd1a-6b857e4d5de0],
Method [GET], Path [/dashboard/widget_chart_data/100] Action: widget_chart_data
[----] I, [2023-01-27T10:02:39.993080 #17089:66d00]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [1a7b7ec1-8d37-44ee-bd1a-6b857e4d5de0],
Method [GET], Path [/dashboard/widget_chart_data/100] Features checked: dashboard_view
[----] I, [2023-01-27T10:02:40.183948 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [33b791e5-2ba9-4493-af46-4a66e91978eb],
Method [GET], Path [/dashboard/widget_report_data/20] Action: widget_report_data
```

```
[----] I, [2023-01-27T10:02:40.184786 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [33b791e5-2ba9-4493-af46-4a66e91978eb],
Method [GET], Path [/dashboard/widget_report_data/20] Features checked: dashboard_view
[----] I, [2023-01-27T10:02:40.395302 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [37b40523-d8e0-4db2-8450-e1d20947f5c9],
Method [GET], Path [/dashboard/widget_chart_data/6] Action: widget_chart_data
[----] I, [2023-01-27T10:02:40.395566 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [37b40523-d8e0-4db2-8450-e1d20947f5c9],
Method [GET], Path [/dashboard/widget_chart_data/6] Features checked: dashboard_view
[----] I, [2023-01-27T10:02:40.441604 #17089:66d00]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [7615b645-1426-49c2-bede-cc6ab2ff0043],
Method [GET], Path [/dashboard/widget_chart_data/74] Action: widget_chart_data
[----] I, [2023-01-27T10:02:40.442043 #17089:66d00]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [7615b645-1426-49c2-bede-cc6ab2ff0043],
Method [GET], Path [/dashboard/widget_chart_data/74] Features checked: dashboard_view
[----] I, [2023-01-27T10:02:40.665204 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [dd1cbac8-db78-4026-9582-3c8eb227d01f],
Method [GET], Path [/dashboard/widget_report_data/16] Action: widget_report_data
[----] I, [2023-01-27T10:02:40.666085 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [dd1cbac8-db78-4026-9582-3c8eb227d01f],
Method [GET], Path [/dashboard/widget_report_data/16] Features checked: dashboard_view
[----] I, [2023-01-27T10:02:41.652177 #17089:66d00]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [8cecb5c5-13da-4404-b6c5-1a0bec224cfb],
Method [GET], Path [/dashboard/widget_report_data/18] Action: widget_report_data
[----] I, [2023-01-27T10:02:41.652878 #17089:66d00]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [8cecb5c5-13da-4404-b6c5-1a0bec224cfb],
Method [GET], Path [/dashboard/widget_report_data/18] Features checked: dashboard_view
[----] I, [2023-01-27T10:07:21.677049 #17089:66d00]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [16bfe549-1d36-49bc-9ce5-6ec260c91021],
Method [GET], Path [/ems_configuration/show_list] Action: show_list
[----] I, [2023-01-27T10:07:25.466777 #17089:66d00]  INFO -- audit: <AuditSuccess> Username
[admin], Role [EvmRole-super_administrator], Request [44f4f937-d391-4891-846e-95cf52939b47],
Method [POST], Path [/ems_configuration/report_data] Action: report_data
[----] I, [2023-01-27T10:07:25.888963 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username
[admin], from: [User.logoff], User admin has logged off
[----] I, [2023-01-27T10:07:31.609288 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username
[joe], from: [Base.audit_success], User joe successfully validated by EVM
[----] I, [2023-01-27T10:07:31.612001 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username
[joe], from: [Base.audit_success], Authentication successful for user joe
[----] I, [2023-01-27T10:07:31.765188 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [39a967af-c46c-47ba-9e5b-1b22489593fc], Method [GET], Path
[/dashboard/show] Action: show
[----] I, [2023-01-27T10:07:31.765429 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [39a967af-c46c-47ba-9e5b-1b22489593fc], Method [GET], Path
[/dashboard/show] Features checked: dashboard_view
[----] I, [2023-01-27T10:07:33.210210 #17089:66d00]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [59e3e495-6d25-4400-aff3-edbe48259f4a], Method [GET], Path
[/dashboard/widget_report_data/19] Action: widget_report_data
[----] I, [2023-01-27T10:07:33.210718 #17089:66d00]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [59e3e495-6d25-4400-aff3-edbe48259f4a], Method [GET], Path
[/dashboard/widget_report_data/19] Features checked: dashboard_view
[----] I, [2023-01-27T10:07:33.272283 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [c25464a9-76f0-495e-a063-af52e727fddf], Method [GET], Path
[/dashboard/widget_report_data/17] Action: widget_report_data
[----] I, [2023-01-27T10:07:33.272675 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [c25464a9-76f0-495e-a063-af52e727fddf], Method [GET], Path
[/dashboard/widget_report_data/17] Features checked: dashboard_view
[----] I, [2023-01-27T10:07:33.368293 #17089:5a398]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [2009ddce-eaa3-46fb-bc92-ea5fd67b3160], Method [GET], Path
[/dashboard/widget_chart_data/6] Action: widget_chart_data
[----] I, [2023-01-27T10:07:33.368664 #17089:5a398]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [2009ddce-eaa3-46fb-bc92-ea5fd67b3160], Method [GET], Path
[/dashboard/widget_chart_data/6] Features checked: dashboard_view
[----] I, [2023-01-27T10:07:33.520539 #17089:13e0c]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [4f942385-cdae-448f-98a4-5ad34f2f7e2d], Method [GET], Path
[/dashboard/widget_chart_data/5] Action: widget_chart_data
[----] I, [2023-01-27T10:07:33.521192 #17089:13e0c]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [4f942385-cdae-448f-98a4-5ad34f2f7e2d], Method [GET], Path
[/dashboard/widget_chart_data/5] Features checked: dashboard_view
[----] I, [2023-01-27T10:07:33.617704 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [6d4f162d-015a-4dc1-ad3b-9359b240e48d], Method [GET], Path
[/dashboard/widget_chart_data/1] Action: widget_chart_data
```

```
[----] I, [2023-01-27T10:07:33.618119 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [6d4f162d-015a-4dc1-ad3b-9359b240e48d], Method [GET], Path
[/dashboard/widget_chart_data/1] Features checked: dashboard_view
[----] I, [2023-01-27T10:07:33.728720 #17089:66d00]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [b9f522e3-d590-4c27-8792-fabfb6b9ff11], Method [GET], Path
[/dashboard/widget_report_data/20] Action: widget_report_data
[----] I, [2023-01-27T10:07:33.729039 #17089:66d00]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [b9f522e3-d590-4c27-8792-fabfb6b9ff11], Method [GET], Path
[/dashboard/widget_report_data/20] Features checked: dashboard_view
[----] I, [2023-01-27T10:07:33.817352 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [f12ff6f5-3c67-485a-8751-5526e33d40ca], Method [GET], Path
[/dashboard/widget_report_data/18] Action: widget_report_data
[----] I, [2023-01-27T10:07:33.818016 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [f12ff6f5-3c67-485a-8751-5526e33d40ca], Method [GET], Path
[/dashboard/widget_report_data/18] Features checked: dashboard_view
[----] I, [2023-01-27T10:07:33.924635 #17089:13e0c]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [5ee0792a-4262-4184-96d0-49c4dd7a2c13], Method [GET], Path
[/dashboard/widget_report_data/16] Action: widget_report_data
[----] I, [2023-01-27T10:07:33.924891 #17089:13e0c]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [5ee0792a-4262-4184-96d0-49c4dd7a2c13], Method [GET], Path
[/dashboard/widget_report_data/16] Features checked: dashboard_view
[----] W, [2023-01-27T10:07:41.826427 #17089:5a5dc]  WARN -- audit: <AuditFailure> Username
[joe], Role [EvmRole-user], Request [fbb15be3-4137-4f12-b10c-6ba4ab451652], Method [GET], Path
[/ops/explorer] Action: explorer
[----] W, [2023-01-27T10:07:41.964517 #17089:5a5dc]  WARN -- audit: <AuditFailure> Username
[joe], Role [EvmRole-user], Request [75430ae8-c62b-40f3-a0f9-fb8b1e43b611], Method [GET], Path
[/dashboard/auth_error] Authentication Error Redirect
[----] I, [2023-01-27T10:07:55.728336 #17089:66cb0]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [8a36b777-e757-41d1-b90a-507fa97ab767], Method [GET], Path
[/report/explorer] Action: explorer
[----] I, [2023-01-27T10:07:57.033934 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [919c4db0-ed64-4075-bc72-e22e85db77f5], Method [POST],
Path [/report/tree_select?id=root&text=All%2520Saved%2520Reports] Action: tree_select
[----] I, [2023-01-27T10:07:57.035044 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [919c4db0-ed64-4075-bc72-e22e85db77f5], Method [POST],
Path [/report/tree_select?id=root&text=All%2520Saved%2520Reports] Features checked:
miq_report_saved_reports
[----] I, [2023-01-27T10:07:57.476854 #17089:5a5dc]  INFO -- audit: <AuditSuccess> Username
[joe], Role [EvmRole-user], Request [833b8d0a-d8e5-4788-98d6-d7361a1c277a], Method [POST],
Path [/report/report_data] Action: report_data
[----] I, [2023-01-27T10:07:59.567570 #17089:5a398]  INFO -- audit: <AuditSuccess> Username
[joe], from: [User.logoff], User joe has logged off
[----] W, [2023-01-27T10:10:34.973016 #17089:66cb0]  WARN -- audit: <AuditFailure> Username
[], Role [], Request [710978e7-20ec-4709-8678-5ea03718eb43], Method [GET], Path
[/ops/explorer] Invalid Session
```

Enable Auditing for Infrastructure Automation Managed services

You can enable auditing for Infrastructure Automation Managed services.

1. Enable auditing during installation or post-installation by updating the **iaconfig** configuration. You can update the configuration by using the following command.

   ```
   oc -n cp4waiops edit iaconfig
   ```

   - Find the **spec** and add **manageservice** to its **spec** section. For example:

     ```
     spec:
       infraAutoComposableComponents:
        name: ibm-management-cam-install
        spec:
          manageservice:
          global:
           audit: true
           license:
             accept: true
     ```

```
  license:
    accept: true
```

2. To configure the IBM Cloud Pak foundational services for receiving audit logs, see Architecture for Audit logging version 3.7.0.

Security and Privacy by Design (SPbD)

Security and Privacy by Design (SPbD) at IBM® is an agile set of focused security and privacy practices, including threat models, privacy assessments, security testing, and vulnerability management.

This document is intended to help you in your preparations for GDPR readiness. It provides information about features of IBM Cloud Pak® for AIOps that you can configure, and aspects of the product's use, to consider for GDPR readiness. This information is not an exhaustive list, due to the many ways that customers can choose and configure features, and the product can be used in itself and with third-party applications and systems.

IBM developed a set of SPbD processes and tools that are used by all of its business units. For more information about the IBM Secure Engineering Framework (SEF) and SPbD, see the IBM Redbooks Security in Development - The IBM Secure Engineering Framework (available in PDF format).

IBM also provides information about the features of IBM® Netcool® Operations Insight® that you can configure, how to use the product securely, and what to consider to help your organization with GDPR readiness. For more information, see Platform considerations for GDPR readiness.

For information about container security, see the Red Hat® documentation: Security and compliance

Encryption in motion

To ensure that encryption in motion is enabled between each service or node within your Red Hat® OpenShift® Container Platform cluster, you must enable IPsec on Red Hat OpenShift Container Platform. When you install Red Hat OpenShift Container Platform, specify an empty object for the **ipsecConfig** parameter to enable IPsec encryption, as in the following example:

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  defaultNetwork:
    type: OVNKubernetes
    ovnKubernetesConfig:
      ipsecConfig: {}
```

Note: IPsec enablement status cannot be changed after cluster installation.

For more information, see IPsec encryption configuration and Specifying advanced network configuration in the Red Hat OpenShift Container Platform documentation.

Platform considerations for GDPR readiness

Notice

This document is intended to help you in your preparations for GDPR readiness. It provides information about features of IBM Cloud Pak® for AIOps that you can configure, and aspects of the product's use, that you should consider to help your organization with GDPR readiness. This information is not an exhaustive list, due to the many ways that clients can choose and configure features, and the large variety of ways that the product can be used in itself and with third-party applications and systems.

Clients are responsible for ensuring their own compliance with various laws and regulations, including the European Union General Data Protection Regulation. Clients are solely responsible for obtaining advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulations that may affect the clients' business and any actions the clients may need to take to comply with such laws and regulations.

The products, services, and other capabilities described are not suitable for all client situations and may have restricted availability. IBM® does not provide legal, accounting, or auditing advice or represent or warrant that its services or products will ensure that clients are in compliance with any law or regulation.

Table of Contents

- GDPR
- Product Configuration for GDPR
- Data Life Cycle
- Data Collection
- Data Storage
- Data Access
- Data Processing
- Data Deletion
- Data Monitoring
- Capability for Restricting Use of Personal Data

GDPR

General Data Protection Regulation (GDPR) has been adopted by the European Union ("EU") and applies from May 25, 2018.

Why is GDPR important?

GDPR establishes a stronger data protection regulatory framework for processing of personal data of individuals. GDPR brings:

- New and enhanced rights for individuals
- Widened definition of personal data
- New obligations for processors
- Potential for significant financial penalties for non-compliance
- Compulsory data breach notification

Read more about GDPR

- ibm.com/GDPR website

Product Configuration - considerations for GDPR Readiness

The following sections describe aspects of data management within IBM Cloud Pak for AIOps and provide information on capabilities to help customers with GDPR requirements.

Data Life Cycle

IBM Cloud Pak for AIOps is an AIOps platform that deploys advanced, explainable AI across the ITOps toolchain so you can confidently assess, diagnose, and resolve incidents across mission-critical workloads.

IBM Cloud Pak for AIOps deals primarily with technical data, some of which might be subject to GDPR. IBM Cloud Pak for AIOps also deals with information about users who manage the deployment. This data is described throughout this document for the awareness of customers responsible for meeting GDPR requirements. This data is persisted on local or remote file systems as configuration files or in databases. Applications that integrate with IBM Cloud Pak for AIOps might deal with other forms of personal data subject to GDPR. The mechanisms that are used to protect and manage data are also available to applications that integrate with IBM Cloud Pak for AIOps. Extra mechanisms might be required to manage and protect personal data that is collected by these applications.

To best understand IBM Cloud Pak for AIOps and its data flows, you must understand how Kubernetes and Docker work. These open source components are fundamental to IBM Cloud Pak for AIOps.

IBM Cloud Pak for AIOps includes a catalog of containerized software and services from IBM® in the default IBM Cloud Pak for AIOps repository list. To view a list of all the IBM Cloud Pak for AIOps charts, see IBM/charts. For considerations about GDPR for the products in the catalog, consult the documentation for those products. Some of the applications available in the catalog are open source software. It is the customer's responsibility to determine and implement any appropriate GDPR controls for open source software. Information on these packages is included in the catalog entry.

What types of data flow through IBM Cloud Pak for AIOps platform

IBM Cloud Pak for AIOps deals with several categories of technical data that might be considered as personal data. Categories include administrator user IDs and passwords, service user IDs and passwords, IP addresses, and Kubernetes node names. IBM Cloud Pak for AIOps also deals with information about users who manage the deployment. Integrated applications might introduce other categories of personal data unknown to IBM Cloud Pak for AIOps.

Information on how this technical data is collected or created, stored, accessed, secured, logged, and deleted is described in later sections of this document.

Personal data used for online contact with IBM

Customers can submit online comments/feedback/requests to contact IBM about IBM Cloud Pak for AIOps subjects in various ways, primarily:

- The public IBM Cloud Pak for AIOps Slack Community
- Public comments area on pages of IBM Cloud Pak for AIOps product documentation in the IBM Documentation

Typically, only the customer name and email address are used to enable personal replies for the subject of the contact. The use of personal data conforms to the IBM Online Privacy Statement.

Data Collection

IBM Cloud Pak for AIOps does not collect any special categories of personal data. It does create and manage technical data, such as an administrator user ID and password, service user IDs and passwords, IP addresses, and Kubernetes node names, which might be considered personal data. IBM Cloud Pak for AIOps also deals with information about users who manage the offering. All such information is only accessible by the administrator.

Applications that run on IBM Cloud Pak for AIOps might collect personal data.

To assess the use of IBM Cloud Pak for AIOps running containerized applications and your need to meet the requirements of GDPR, you must consider the types of personal data that the application collects and aspects of how that data is managed, such as:

- How is the data protected as it flows to and from the application? Is the data encrypted in transit?
- How is the data stored by the application? Is the data encrypted at rest?
- How are credentials, which are used to access the application, collected and stored?
- How are credentials, which are used by the application to access data sources, collected and stored?
- How is data collected by the application removed as needed?

This list is not a definitive list of the types of data collected by IBM Cloud Pak for AIOps. It is provided as an example for consideration. If you have any questions about the types of data, contact IBM.

Types of personal data

- Basic Personal Information (such as name, address, phone number, email)
- Technically Identifiable Personal Information (such as device IDs, usage-based identifiers, static IP addresses - when linked to an individual).

Special categories of personal data

- IBM Cloud Pak for AIOps was not designed to process any special categories of personal data.

Data storage

IBM Cloud Pak for AIOps persists technical data in stateful stores on local or remote file systems as configuration files or in databases. Consideration must be given to securing all data at rest. IBM Cloud Pak for AIOps supports encryption of data at rest in stateful stores. For more information, see Security considerations

Data access

IBM Cloud Pak for AIOps provides a number of groups and roles for controlling data access. The groups and roles enable differentiation between normal users and those with extra privileges. For more information, see Roles and permissions.

Data Processing

In general, data that is used for authentication must be in a directory service or LDAP. Databases are provisioned during installation. Make sure to maintain them throughout product lifecycles.

Basic recommendations

- Regularly back up data, according to your business needs and to the risk level.
- Encrypt data backups.
- When data is no longer used, delete the databases or archive them for future use.
- As a data controller, provide means to satisfy data access requests for personal information or other compliance requests.

Further considerations

- Make sure that control of access to databases is in place and effective.
- Use strong credentials.
- Protect the REST administration APIs with proper credentials.
- Use HTTPS or equivalent secure communication protocols for all the connections.
- Remove or change all default passwords.

Data Deletion

Article 17 of the GDPR states that data subjects have the right to request that their personal data is removed from the systems of controllers and processors, without undue delay. Implement appropriate controls and tools to satisfy this right.

IBM Cloud Pak for AIOps does not require any special method for data deletion.

Data that reflects personally identifiable information (PII) can be in all stages of the data processing pipeline. Data deletion must include all these stages. Administrators can use IBM Cloud Pak for AIOps features to remove user data.

Data monitoring

Regularly test, assess, and evaluate the effectiveness of your technical and organizational measures to comply with GDPR. These measures should include ongoing privacy assessments, threat modeling, centralized security logging, and monitoring, among others.

Capability for Restricting Use of Personal Data

Using the facilities summarized in this document, IBM Cloud Pak for AIOps enables a user to restrict usage of any technical data that is considered personal data.

Under GDPR, users have rights to access, modify, and restrict processing. Refer to other sections of this document to manage the following controls:

- Right to access
    - Administrators can use IBM Cloud Pak for AIOps features to provide individuals access to their data.

    - Administrators can use IBM Cloud Pak for AIOps features to provide individuals information about what data IBM Cloud Pak for AIOps platform holds about the individual.
- Right to modify
    - Administrators can use IBM Cloud Pak for AIOps features to allow an individual to modify or correct their data.
    - Administrators can use IBM Cloud Pak for AIOps features to correct an individual's data for them.
- Right to restrict processing
    - Administrators can use IBM Cloud Pak for AIOps features to stop processing an individual's data.

When configuring logs from their environments to be stored or processed in IBM Cloud Pak for AIOps, users are responsible for filtering out personal information or other sensitive information. For more information about filtering logs, see the following topics:

- Creating ELK connections
- Creating Falcon LogScale connections
- Creating LogDNA connections
- Creating Splunk connections

Data retention

IBM Cloud Pak for AIOps retains data for processing, analysis, and historical purposes. Review the following information to learn more about the retention period for each data type and how to configure retention period for different resources.

- Data retention periods
- Configure the retention period for topology resources
- Configure the retention period for log resources

Data retention periods

Retention periods for data

Data type	Retention period
Logs	- Retention period: 14 days - Normalized logs are retained for 14 days.
Metrics	- Retention period: 15 days - Metric time series data is retained for 15 days.
Events	- Retention period: N/A - Events are not retained after processing.
Alerts	- Retention period: 90 days - Open alerts are kept indefinitely. - Closed alerts are archived and retained for 90 days.
Incidents	- Retention period: N/A - Open incidents are kept indefinitely. - Closed incidents are purged when closed.
Tickets	- Retention period: Indefinite - Closed tickets from a historical time frame are retrieved from the ticketing system based on a user defined date range. Each subsequent retrieval adds to the total data set. Retention controls are not provided.
Topology	- Retention period: 30 days - Valid resources are kept indefinitely. - Deleted or changed resources are deleted after 30 days. - Event correlation and associated analytics use the current topology.

Note: The retention periods are determined through careful review and analysis to represent good default values. The retention periods keep sufficient data for learning normal behavior patterns, while allowing for adjustments when the normal behavior changes and for discarding past behavior that is obsolete. You are not recommended to change the retention periods.

Configure the retention period for topology resources.

If required, you can configure the retention period for topology resources. For instance, you can increase the retention period to a maximum of 90 days.

The retention period for historical resource data is configured through the **HISTORY_TTL** time to live (TTL) configuration property. You can increase the retention period by increasing this property value. When a resource is deleted, the resource no longer appears in the UI. Historic representations of the resource, however, are retained and can be accessed in the UI history timeline until the history TTL limit is reached, after which the data is deleted.

Caution: Performance and scalability are affected by the number of resources that are managed, the amount of history present for each resource, and the ingestion rate. If you increase the retention period for historical resource data from 30 days (up to a maximum of 90 days), your system performance (when rendering views) might deteriorate, if as a result of this increase resources have in excess of 25,000 historical entries.

To configure the **HISTORY_TTL** time to live (TTL) property, complete the following steps on your cluster:

1. Log in to your cluster.
2. Open the **$ASM_HOME/.env** file for editing in your preferred editor.
3. Edit the **HISTORY_TTL** setting to set your preferred retention period (in hours). You can change the setting of **720** hours (30 days) to a maximum of **2160** hours (90 days).
4. Restart the topology service:

   ```
   $ASM_HOME/bin/asm_start.sh
   ```

Configure the retention period for log resources.

If required, you can configure the retention period for log resources.

Use the following steps to configure the retention period for log resources:

1. Log in to the Red Hat OpenShift Container Platform cluster.
2. Click **Workloads**.
3. Click **ConfigMaps** > **aimanager-aio-curator-config**.
4. Click **YAML** and edit the value under the **unit_count**.

   ```
   unit_count: 10
   ```

   If you set the value of **unit_count** to **10**, this reduces the retention period to 10 days.

Getting started

Now that you want to get started with IBM Cloud Pak® for AIOps, it's time to learn about all of the capabilities afforded to you by IBM Cloud Pak, and how to take advantage of these capabilities.

Before installing IBM Cloud Pak for AIOps, take the time to understand your needs. IBM Cloud Pak for AIOps includes many features by default. If you have a clear understanding of what you're looking to do with IBM Cloud Pak for AIOps, your installation might be more straightforward.

Ensure that you have a fundamental understanding of the following software, programming languages, and tools:

- Red Hat® OpenShift® Container Platform

  Having a solid understanding of Red Hat OpenShift Container Platform facilitates your installation and administration of IBM Cloud Pak for AIOps. This understanding includes how to browse and use the Red Hat OpenShift Container

Platform web console and catalog, how to install operators, how to run `oc` commands, and how to complete general tasks within Red Hat OpenShift Container Platform.

- Kubernetes operators

  For more information operators, see the Red Hat documentation about Operators.

To help first-time users get started, guided tours are included within the UI console. These tours introduce users to the various UI options and components, and demonstrate how to complete key tasks across the platform. Users can access the tours at any time by clicking the Tours icon in the console toolbar, and selecting their preferred tour. For more information about the available guided tours, see Guided tours.

For more information about working with the IBM Cloud Pak, see the following topics:

- IBM Cloud Pak for AIOps workflows
- Archetypes

Archetypes and user roles

For IBM Cloud Pak® for AIOps, different persona archetypes are defined to help you understand the different available and supported tasks so that you can get your environment up and running or begin your day to day operations quickly and efficiently.

Archetypes are a set of basic characteristics that a user role can fulfill. You can think of an archetype as a templated job description for a particular user role. For example, in IBM Cloud Pak for AIOps, the Site reliability expert is responsible for many tasks assessing and remediating problems. The Site reliability expert falls under the Infrastructure engineer archetype, that is, someone who is focused on keeping the entire IT infrastructure of an organization up and running 24 x 7.

The following defined archetypes and user roles are referenced within the product documentation as the user for completing different use cases and tasks:

- Application developer
- Infrastructure engineer

Archetype: Application developer

Application developers in IBM Cloud Pak for AIOps are responsible for building things outside of the scope of enterprise IT security. Application developers want to solve problems early so that they can focus on new feature development and not dwell on debugging IT issues. Application developers work hand in hand with the Security operations manager and Infrastructure engineer to identify and fix issues. The following user roles are often associated with the Application developer archetype:

- Software engineer
- Software developer
- DevOps engineer
- Full stack developer

The following table lists the high-level key tasks that Application developers can complete with IBM Cloud Pak for AIOps:

Table 1. Archetype: Application developer tasks and references

Tasks	References
Monitor outgoing connections for issues	ChatOps insight management
Debug issues that are associated with applications and related services	Managing incidents

Archetype: Infrastructure engineer

Infrastructure engineers play a vital role in IBM Cloud Pak for AIOps as they ensure that systems are always up and running. They serve as the go-between from the strategy and high level thinking of the Security operations manager to the

implementation by the Application developers. Infrastructure engineers are always aware of their platforms, ensuring that problems are detected and remediated as quickly as possible. If all goes well, the Infrastructure engineer is responsible for a platform that never goes down. The following user roles are often associated with the Infrastructure engineer archetype:

- Site reliability engineer
- Systems engineer
- Operations engineer
- Platform engineer

The following table lists the high-level key tasks that Infrastructure engineers can complete with IBM Cloud Pak for AIOps:

Table 2. Archetypes: Infrastructure engineer tasks and references

Tasks	References
Install IBM Cloud Pak for AIOps	Installing
Administer IBM Cloud Pak for AIOps	Administering IBM Cloud Pak for AIOps
Setting up data and tool connections	Configuring data and tool connections
Training AI models	Configuring AI training
Managing topologies	Application management
Monitor outgoing connections for issues	ChatOps insight management
Debug issues that are associated with applications and related services	Managing incidents

IBM Cloud Pak for AIOps workflows

You decided to install IBM Cloud Pak® for AIOps, now it's time to explore the key workflows of your users and their interaction with IBM Cloud Pak for AIOps.

Understanding how you can accomplish your tasks, who's responsible, and how each workflow fits together gives you a holistic picture of your entire AIOps-infused IT infrastructure management experience. You can bring IBM Cloud Pak for AIOps to life by deploying more operators, customizing your configurations, and optimizing IBM Cloud Pak for AIOps performance. For more information about how to get the most out of your team, see Archetypes.

Installing IBM Cloud Pak for AIOps

Platform engineers prepare all required hardware and software, and then install the Red Hat® OpenShift® Container Platform. They prepare OpenShift clusters and install IBM Cloud Pak for AIOps into those clusters.

For more information about installing IBM Cloud Pak for AIOps, see the following sections:

- Planning installation, configuration, and overall cluster requirements
- Preparing for and installing IBM Cloud Pak for AIOps

Guided tours

To help first-time users get started with some workflows, guided tours are included within the UI console across the different dashboards. These tours introduce users to the various UI options and components, and demonstrate how to complete key tasks across the platform. Users can access the tours at any time by clicking the Tours icon in the console toolbar, and selecting their preferred tour.

For more information about the available guided tours, see Guided tours.

Setting up data and tool connections

You can create connections for logs, to establish a baseline of normal behavior and identify anomalies. Other connections provide event, applications, and infrastructure data to IBM Cloud Pak for AIOps.

Systems engineers create connections to ticketing and to ChatOps tools. The connections to ticketing tools have a dual purpose: they enable ticket data to be collected for change request and similar ticket training. The same connection also enables synchronization between IBM Cloud Pak for AIOps incidents and corresponding ServiceNow tickets. The connections to Microsoft Teams and Slack ChatOps tools enable incident notifications to be pushed to these ChatOps tools. These notifications are for the attention of Site Reliability Engineers who are tasked with resolving those incidents.

For more information about setting up tool connections, see the following sections:

- Defining outgoing connections
- Defining incoming connections

Loading data and training AI

As a prerequisite to training your AI to provide insights on your monitored environment, systems engineers load event, log, metric, ticket, and other data from various data sources. They then run different types of AI training to generate AI models such as log anomaly models, which can easily identify and report on anomalies in your log data. Insights that are generated by these AI models are used to enrich incidents with related alert, similar ticket, and event grouping information.

For more information about loading data and training AI, see the following sections:

- Planning data loading and training
- Defining incoming connections
- Configuring AI training

Building applications

Systems engineers are responsible for building the applications that are monitored by Site Reliability Engineers. Systems engineers build applications from the ground up, by first defining observer connections to import resource topology data, and then defining templates to automatically generate resource groups from the incoming observer streams. When they have an incoming dynamic stream of resource groups, they build business applications that are made up of multiple resource groups.

For more information about building applications, see the following sections:

- Defining observer jobs
- Defining templates with template builder
- Creating and editing applications
- Viewing applications

Creating automations

System administrators set up policies, runbooks, and actions to detect and remediate incoming alerts. Policies are rules that contain a condition and a set of outcomes that can be manual or automated. Runbooks help to solve common operational problems by automating procedures that do not require human interaction. Procedure-based documentation can be transferred into manual runbooks. Actions in runbooks are the collection of several manual steps into a single automated entity. Actions can be added to create semi-automated or fully automated runbooks.

For more information about creating automations, see the following sections:

- Creating policies
- Creating a runbook
- Creating an action

Resolving incidents

Site reliability engineers are responsible for investigating and resolving incidents and incidents that are associated with applications and infrastructure. They use ChatOps tools such as Microsoft Teams and Slack to monitor incident channels, and investigate incidents by using application, topology, ticketing, alert, and similar ticket information provided in the incident notification. They act to resolve the incidents, where necessary involving Applications developers.

For more information about resolving incidents, see the following sections:

- ChatOps insight management
- Application incidents

Administering the cluster

Platform engineers administer the cluster on which IBM Cloud Pak for AIOps is installed on an ongoing basis. They ensure that all nodes, pods, and other elements within the cluster are performing as expected. Using the IBM Cloud Pak Administration panel, they monitor trends, display workload summaries, and display system utility status. They also support the systems engineer with complex cluster administration tasks such as multitenancy management, project (namespace) management, and authorization and authentication of users within the cluster.

For more information about administering the cluster, see the following sections:

- Administration panel

Administering IBM Cloud Pak for AIOps

Systems engineers, or in some organizations, Operations engineers, are responsible for authenticating and managing users. They can also customize the UI experience by setting console branding and customizing what content users see on the home page. For more complex cluster-based tasks such as multitenancy management, project (namespace) management, and authorization and authentication of users within the cluster, they collaborate with the Platform engineer.

For more information about administering IBM Cloud Pak for AIOps, see the following sections:

- Managing user access control
- Customizing the platform UI

Installing

You can install IBM Cloud Pak® for AIOps, Infrastructure Automation, or both. Online and offline deployments are supported.

If you have an existing on-premises IBM Tivoli Netcool/OMNIbus deployment, then you can connect IBM Cloud Pak for AIOps to it with the Netcool connector. For more information, see Creating IBM Tivoli Netcool/OMNIbus connections.

If you have an existing on-premises IBM Tivoli Netcool/Impact deployment, then you can connect IBM Cloud Pak for AIOps to it with the IBM Tivoli Netcool/Impact connector. For more information, see Creating IBM Tivoli Netcool/Impact connections

Before you begin

- IBM Cloud Pak for AIOps and Infrastructure Automation can co-exist on the same cluster, but you cannot have multiple instances of either of them on the same cluster.
- You can optionally verify the IBM Cloud Pak for AIOps images before you install IBM Cloud Pak for AIOps. For more information, see Verifying images.

Installation options

Read about planning an installation, and then select one of the installing topics.

- Planning
- Installing IBM Cloud Pak for AIOps
- Installing Infrastructure Automation

Planning

Review the following information to help you plan your installation.

Supported Red Hat OpenShift versions

Learn about which Red Hat® OpenShift® Container Platform versions you can deploy on. For more information, see Supported OpenShift versions.

Supported cloud platforms

Learn about the public cloud platforms that you can deploy on. For more information, see Supported cloud platforms.

Storage requirements

Review the storage requirements for your deployment, and learn about supported storage providers and storage classes. For more infomation, see Storage requirements.

Hardware requirements

Review the memory and hardware requirements for your deployment. For more information, see Hardware requirements.

Operator installation mode

The installation mode can be **`OwnNamespace`** or **`AllNamespaces`**. For more information, see Operator scope.

High availability

Review information about how to configure high availability (HA) for your deployment. For more information, see High availability (HA) considerations

Security considerations

IBM Cloud Pak for AIOps supports different mechanisms for securing your environment and your data. In addition, as you are planning or going through an installation of IBM Cloud Pak for AIOps, you might need to create or provide certificates. For more information, see Security considerations.

Licensing

As part of using IBM Cloud Pak for AIOps, tracking your license usage is required. Although IBM Cloud Pak for AIOps provides license options not tracked directly by the product, to track usage, you must deploy the IBM Cloud Pak foundational services License Service on all clusters where IBM Cloud Pak for AIOps is installed. This background service collects and stores the license usage information for tracking license consumption and for audit purposes. For more information, Licensing.

Supported Red Hat OpenShift Container Platform versions

IBM Cloud Pak for AIOps supports and recommends installing on a supported Red Hat® OpenShift® Container Platform EUS minor version for the duration of its support period. IBM Cloud Pak for AIOps adheres to the IBM Cloud Pak for AIOps and Red Hat OpenShift version compatibility objective. For more information about this objective, see IBM Cloud Pak and Red Hat OpenShift version compatibility objective.

For more information about the support for Red Hat OpenShift versions, see Red Hat OpenShift Container Platform Life Cycle Policy.

Notes:

- The hardware architecture for installing IBM Cloud Pak for AIOps must be AMD64.
- If you are planning to upgrade to Red Hat OpenShift 4.12 as part of an upgrade to IBM Cloud Pak for AIOps 4.2.0, you must complete the IBM Cloud Pak for AIOps upgrade before you upgrade to Red Hat OpenShift 4.12.

Supported OpenShift versions

	Platform	Supported Red Hat OpenShift Container Platform EUS version
IBM Cloud Pak for AIOps	Linux x86_64	4.12 (minimum 4.12.50)
Infrastructure Automation (Installed with IBM Cloud Pak for AIOps)	Linux x86_64	4.12 (minimum 4.12.50)
Infrastructure Automation (Installed stand-alone)	Linux x86_64 Linux on Power (ppc64le) Linux® on IBM Z® and LinuxONE (s390x)	4.12 (minimum 4.12.50)

Supported cloud platforms

IBM Cloud Pak® for AIOps and Infrastructure Automation are supported on the following public cloud platforms.

For more information about storage providers that you can use on these platforms, see Recommended storage providers.

Amazon Web Services (AWS)

You can install IBM Cloud Pak for AIOps and Infrastructure Automation on AWS with:

- Red Hat® OpenShift®
- Red Hat OpenShift Service on AWS (ROSA). For more information about ROSA, see the Red Hat documentation Red Hat OpenShift Service on AWS.

  For more information about installing IBM Cloud Pak for AIOps on ROSA, see Installing IBM Cloud Pak for AIOps on ROSA.

Google Cloud Platform (GCP)

You can install on GCP with Red Hat OpenShift. For more information about installing Red Hat OpenShift on GCP, see the Red Hat OpenShift documentation Installing on GCP.

For more information about IBM Cloud Pak for AIOps on GCP, see Installing IBM Cloud Pak for AIOps on GCP.

Note: Infrastructure Automation is not supported on GCP.

IBM Cloud

You can install on IBM Cloud® with:

- Red Hat OpenShift
- Red Hat® OpenShift® on IBM Cloud® service (ROKS). For more information about ROKS, see the Red Hat OpenShift documentation Red Hat OpenShift on IBM Cloud.

  For more information about installing IBM Cloud Pak for AIOps on ROKS, see Installing IBM Cloud Pak for AIOps on ROKS.

Microsoft® Azure

You can install on Azure with:

- Red Hat OpenShift
- Azure Red Hat OpenShift (ARO). For more information about ARO, see the Microsoft documentation Azure Red Hat OpenShift.

  For more information about installing IBM Cloud Pak for AIOps on (ARO), see Installing IBM Cloud Pak for AIOps on ARO.

Storage providers and requirements

You need persistent storage configured for your cluster to install and use IBM Cloud Pak® for AIOps and Infrastructure Automation.

Storage requirements

IBM Cloud Pak for AIOps and Infrastructure Automation require persistent storage that supports the RWO (read-write-once) and RWX (read-write-many) access modes.

For information about the persistent volume and sizing requirements, see Hardware requirements.

Federal Information Processing Standards (FIPS) storage requirements: If your environment must support FIPS, then you must enable FIPS support when you install IBM Cloud Pak for AIOps or Infrastructure Automation. For more information, see Federal Information Processing Standards (FIPS).

Recommended storage providers

The following table shows the tested and supported storage providers that are recommended for a deployment of IBM Cloud Pak for AIOps or Infrastructure Automation.

Table 1. Supported storage providers

Platform	IBM Cloud Storage (Block and File)	Red Hat® OpenShift® Data Foundation	IBM Storage Fusion	IBM Spectrum Scale Container Native	Portworx
Azure Red Hat OpenShift (ARO)		Yes			Yes
Google Cloud Platform (GCP)		Yes			Yes
Red Hat OpenShift Container Platform		Yes	Yes	Yes	Yes
Red Hat OpenShift on IBM Cloud (ROKS)	Yes	Yes			
Red Hat OpenShift Service on AWS (ROSA)					Yes

Notes:

- IBM Spectrum Scale Container Native and Red Hat® OpenShift® Data Foundation are part of IBM Storage Suite for IBM Cloud Paks.
- IBM Cloud Pak for AIOps requires persistent RWX (read-write-many) storage. Red Hat® does not currently support Red Hat® OpenShift® Data Foundation (ODF) on ROSA. Portworx is the only recommended storage provider that provides a RWX storage solution for a deployment of IBM Cloud Pak for AIOps on ROSA. Portworx is available for a free 30 day trial, but will require a license for a longer timespan and production usage.

Other storage providers

The preceding storage providers are the only providers that are tested and validated for a deployment of IBM Cloud Pak for AIOps or Infrastructure Automation. You can choose to use an alternate storage provider if they meet the requirements for

deploying IBM Cloud Pak for AIOps or Infrastructure Automation.

Your chosen storage provider must meet the same storage and hardware requirements as the recommended storage providers. For instance, for deploying IBM Cloud Pak for AIOps, your chosen provider must support the required access modes and storage modes.

Note: If you choose to use an alternate storage provider, your overall performance can differ from any listed sizings, throughput rates or other performance metrics that are listed in the IBM Cloud Pak for AIOps documentation. Work with your IBM Sales representative to ensure that your chosen storage provider is sufficient for your deployment plan.

Storage classes for recommended storage providers

If you choose to use one of the recommended storage providers, you need to configure specific storage classes, which are identified in the following table. Then, when you are installing IBM Cloud Pak for AIOps, you need to specify the appropriate storage class for your chosen storage provider.

Table 2. Supported storage providers

Storage provider	storage_class_name	large_block_storage_class_name
IBM Cloud®	ibmc-file-gold-gid	ibmc-block-gold
Red Hat® OpenShift® Data Foundation	ocs-storagecluster-cephfs	ocs-storagecluster-ceph-rbd
IBM Storage Fusion	ibm-spectrum-scale-sc	ibm-spectrum-scale-sc
IBM Spectrum Scale Container Native	ibm-spectrum-scale-sc	ibm-spectrum-scale-sc
Portworx	portworx-fs	portworx-aiops

Storage class requirements

For production environments, storage classes must have **`allowVolumeExpansion`** enabled. This allows persistent volumes to be expanded if necessary, to avoid storage from filling up and causing unrecoverable failures. This is also highly recommended for starter deployments, since without it you are limited to the default capacity that might not be sufficient for your specific needs.

To enable **`allowVolumeExpansion`**, edit the storage class to enable expansion. Follow the instructions in the Red Hat documentation Enabling volume expansion support.

Installing recommended storage providers and configuring storage classes

Learn more about installing a recommended storage providers and configuring storage.

IBM Cloud File Storage and IBM Cloud Block Storage

For more information about IBM Cloud storage, see Storing data on classic IBM Cloud File Storage and Storing data on classic IBM Cloud Block Storage in the IBM Cloud Docs.

Installing IBM Cloud File Storage and IBM Cloud Block Storage

The storage classes that are required by IBM Cloud Pak for AIOps are created when Red Hat OpenShift on IBM Cloud (ROKS) is installed.

Red Hat OpenShift Data Foundation (ODF)

Red Hat OpenShift Data Foundation is available for purchase through the IBM Storage Suite for IBM Cloud Paks. Red Hat OpenShift Data Foundation is an implementation of the open source Ceph storage software, which is engineered to provide data and storage services on Red Hat OpenShift. Use version 4.12 or higher.

Installing Red Hat OpenShift Data Foundation

For more information about deploying Red Hat OpenShift Data Foundation, see Deploying OpenShift Data Foundation in the Red Hat documentation. Choose the appropriate deployment instructions for your deployment platform.

The following storage classes are installed when you deploy Red Hat OpenShift Data Foundation:

- **`ocs-storagecluster-ceph-rbd`**

- `ocs-storagecluster-cephfs`
- `openshift-storage.noobaa.io`
- `ocs-storagecluster-ceph-rgw`

Important: Red Hat OpenShift Data Foundation provide additional configuration steps, which can alleviate potential problems with delayed space reclamation. For more information, see Reclaiming space on target volumes in the Red Hat OpenShift Data Foundation documentation.

These additional configuration steps are not specific to IBM Cloud Pak for AIOps, and are required for any application that experiences a high volume of writes that are then deleted. For IBM Cloud Pak for AIOps, this might be an event storm where large numbers of events are ingested and resolved, or when a large quantity of short-term metric data is ingested and then deleted. Failure to properly configure storage can cause out of disk space errors that might not be readily apparent. In some scenarios, you might need to contact your storage provider to ensure proper storage configuration for your needs.

IBM Storage Fusion

IBM Storage Fusion is a container-native hybrid cloud data platform that offers simplified deployment and data management for Kubernetes applications on Red Hat OpenShift Container Platform. IBM Storage Fusion is designed to meet the storage requirements of modern, stateful Kubernetes applications and to make it easy to deploy and manage container-native applications and their data on Red Hat OpenShift Container Platform. For more information, see the IBM Fusion documentation.

Installing IBM Storage Fusion

When using IBM Cloud Pak for AIOps with IBM Storage Fusion, ensure that your version of Red Hat OpenShift Container Platform is supported for your version of IBM Storage Fusion. Before you install IBM Storage Fusion, ensure that you meet all of the prerequisites. For more information, see IBM Storage Fusion Prerequisites.

For more information about deploying IBM Storage Fusion, see Deploying IBM Storage Fusion.

IBM Spectrum Scale Container Native

IBM Spectrum Scale Container Native is available for purchase through the IBM Storage Suite for IBM Cloud Paks. IBM Spectrum Scale is a cluster file system that provides concurrent access to a single file system or set of file systems from multiple nodes. The nodes can be SAN-attached, network attached, a mixture of SAN-attached and network attached, or in a shared nothing cluster configuration. This enables high-performance access to this common set of data to support a scale-out solution or to provide a high availability platform. IBM Spectrum Scale Container Native Storage Access must be at version 5.1.1.3 or higher, with IBM Spectrum Scale Container Storage Interface version 2.3.0 or higher.

For more information, see the IBM Spectrum Scale Container Native documentation.

Requirements

To use IBM Spectrum Scale Container Native with IBM Cloud Pak for AIOps, you do not require a separate license. You can use up to 12 TB of IBM Spectrum Scale Container Native storage for up to 36 months, fully supported by IBM, within your production environments (Level 1 and Level 2). If you exceed these terms, a separate license is required.

To install and use IBM Spectrum Scale Container Native, your cluster must meet the requirements in the IBM Spectrum Scale Container Native documentation.

Installing IBM Spectrum Scale Container Native

To install IBM Spectrum Scale Container Native and the IBM Spectrum Scale Container Storage Interface, follow the IBM Spectrum Scale Container Native installation documentation.

When IBM Spectrum Scale Container Native is installed, the **`ibm-spectrum-scale-sc`** storage class is installed. This class is used as both the ReadWriteMany storage class and large block storage class when you are installing IBM Cloud Pak for AIOps. This StorageClass includes a parameter to set permissions of data within the StorageClass to *777*, which is required to support the Kubernetes SubPath feature. For more information about the permissions field for the class, see the IBM Spectrum Scale CSI Driver documentation.

If you encounter any errors or issues with installing or using IBM Spectrum Scale Container Native or the IBM Spectrum Scale Container Storage Interface, refer to the following documentation:

- IBM Spectrum Scale Container Native
- IBM Spectrum Scale Container Storage Interface

Portworx

Important: `sharedv4` volumes require NFS ports to be open. For more information, see the Portworx documentation Open NFS Ports.

Installing Portworx

1. Install the Portworx operator and configure a Portworx StorageCluster.

   Two editions of Portworx are available: Portworx Enterprise and Portworx Essentials. Portworx Enterprise is suitable for production deployments. Portworx Essentials is suitable only for demonstration deployments, as it can be used only on clusters of five nodes or less, and includes storage size limits. For more information, see the Portworx documentation Installing Portworx on Openshift and Deploying Portworx using the Operator.

   Note: You must be a cluster administrator to install Portworx on the cluster.

2. Define two storage classes: `portworx-fs` and `portworx-aiops`.

   The storage classes are scoped to the cluster, so setting the project (namespace) is not required. Portworx recommends a replication factor (parameters.repl) of 2 or 3. For more information, see the Portworx documentation Running in Production.

   You can define the storage classes with either of the following methods:

   - Option 1: Define the Portworx storage classes with the OpenShift console
   - Option 2: Define the Portworx storage classes with the OpenShift CLI

   Option 1: Define the Portworx storage classes with the OpenShift console:

   1. Log in to your OpenShift cluster's console.
   2. Create the portworx-fs storage class. This storage class is used for other components, such as IBM Cloud Pak foundational services.

      Click the plus icon on the upper right to open the **Import YAML** dialog box, paste in the following content, and then click **Create**.

      ```
      kind: StorageClass
      apiVersion: storage.k8s.io/v1
      metadata:
        name: portworx-fs
      provisioner: kubernetes.io/portworx-volume
      parameters:
        repl: "3"
        io_profile: "db"
        priority_io: "high"
        sharedv4: "true"
      allowVolumeExpansion: true
      ```

   3. Create the portworx-aiops storage class.

      Click the plus icon on the upper right to open the Import YAML dialog box, paste in the following content, and then click Create.

      ```
      kind: StorageClass
      apiVersion: storage.k8s.io/v1
      metadata:
        name: portworx-aiops
      provisioner: kubernetes.io/portworx-volume
      parameters:
        repl: "3"
        priority_io: "high"
        snap_interval: "0"
        io_profile: "db"
        block_size: "64k"
      allowVolumeExpansion: true
      ```

Option 2: Define the Portworx storage classes with the OpenShift CLI:

Run the following commands to configure the cluster with the following storage class resources.

1. Create the portworx-fs storage class.

```
cat << EOF | oc apply -f -
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: portworx-fs
provisioner: kubernetes.io/portworx-volume
parameters:
  repl: "3"
  io_profile: "db"
  priority_io: "high"
  sharedv4: "true"
allowVolumeExpansion: true
EOF
```

2. Create the portworx-aiops storage class.

```
cat << EOF | oc apply -f -
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: portworx-aiops
provisioner: kubernetes.io/portworx-volume
parameters:
  repl: "3"
  priority_io: "high"
  snap_interval: "0"
  io_profile: "db"
  block_size: "64k"
allowVolumeExpansion: true
EOF
```

Hardware requirements

Learn about the hardware requirements for a deployment of IBM Cloud Pak® for AIOps on Red Hat® OpenShift® Container Platform. Your hardware must be able to support Red Hat OpenShift, IBM Cloud Pak for AIOps, and your chosen storage solution.

You can deploy a starter, production, or custom sized deployment of IBM Cloud Pak for AIOps. The processing abilities and hardware requirements vary for each deployment size.

Important:

- You cannot change your selected deployment size after installation.
- Multi-zone high availability disaster resolution (HADR) is available as a nonproduction technology preview. For more information, see Installing IBM Cloud Pak for AIOps on a multi-zone architecture (multi-zone HADR).
- vCPU is defined as when one x86 CPU splits each of its physical cores into virtual cores (vCPU). It is assumed that one x86 CPU's physical core can be split into two logical vCPUs.
- If Red Hat OpenShift is installed on VMware virtual machines, set the value of the `sched.cpu.latencySensitivity` parameter to high.

Review the information in the following sections:

- Hardware requirements - Red Hat OpenShift only
- Hardware requirements - IBM Cloud Pak for AIOps
- Hardware requirements - Storage
- IOPS requirements
- Processing abilities
- Custom sizing
- Adjusting PVC sizing

- Requests and limits
- Infrastructure Automation

Hardware requirements - Red Hat OpenShift only

A Red Hat OpenShift cluster has master nodes and worker nodes. The following tables show the minimum node requirements for Red Hat OpenShift to withstand the demands of starter and production deployments of IBM Cloud Pak for AIOps. IBM Cloud Pak for AIOps has many Kubernetes operators that interface with the `API server` and `etcd` storage, and so master nodes must be adequately sized. Each Red Hat OpenShift worker node must meet the minimum size requirement to accommodate the placement of larger IBM Cloud Pak for AIOps pods.

IBM Cloud Pak for AIOps requires your Red Hat OpenShift cluster to have at least three master nodes. The number of worker nodes that are required varies depending on the storage and processing requirements of your IBM Cloud Pak for AIOps deployment.

Table. Master node minimum requirements

Resource	Starter	Production
Master node count	3	3
vCPU per node	4	4
Memory (GB) per node	16	16
Disk per node	120	120

Table. Worker node minimum requirements

Resource	Starter	Production
vCPU per node	8	8
Memory (GB) per node	12	20
Disk per node	120	120

Note: Table 2 gives the absolute minimum requirements that each worker node must meet. The size of each of your worker nodes can vary, but the combined resources of your worker nodes must comply with the resource totals in Table 3.

Hardware requirements - IBM Cloud Pak for AIOps

The following table gives the hardware requirements for starter and production deployments of IBM Cloud Pak for AIOps. The resource requirements are given for the following scenarios:

- IBM Cloud Pak for AIOps only - you are deploying IBM Cloud Pak for AIOps on an existing Red Hat OpenShift cluster.
- IBM Cloud Pak for AIOps and Red Hat OpenShift - you are creating a new Red Hat OpenShift cluster and deploying IBM Cloud Pak for AIOps onto it.

Table. Minimum requirements for starter and production deployments

Resource	IBM Cloud Pak for AIOps only (Starter)	IBM Cloud Pak for AIOps and Red Hat OpenShift (Starter)	IBM Cloud Pak for AIOps only (Production)	IBM Cloud Pak for AIOps and Red Hat OpenShift (Production)
Master node count	•	3	•	3
Worker node count	3	3	10	10
Total vCPU	62	74	162	174
Total memory (GB)	140	188	372	420

The Red Hat OpenShift master and worker nodes must meet the minimum size requirements in Hardware requirements - Red Hat OpenShift only.

Important: These values do not include CPU and memory resources for hosting a storage provider, such as Red Hat® OpenShift® Data Foundation or Portworx. Storage providers can require more nodes, more resources, or both, to run. The extra resources that are needed can vary based on your selected storage provider. A general recommendation is for one extra

worker node for a starter deployment, and for three extra worker nodes for a production deployment. Consult your storage provider's documentation for exact requirements.

In addition to the default starter and production size deployments, you can choose to deploy a custom-sized deployment of IBM Cloud Pak for AIOps. For more information about customizing the size of your IBM Cloud Pak for AIOps deployment, see Custom sizing.

Hardware requirements - Storage

IBM Cloud Pak for AIOps requires persistent storage. Red Hat OpenShift uses the Kubernetes persistent volume (PV) framework. PVs are storage resources in the cluster, and persistent volume claims (PVCs) are storage requests that are made on those PVs by IBM Cloud Pak for AIOps. For more information about persistent storage in Red Hat OpenShift clusters, see Understanding persistent storage.

For more information about setting up storage for IBM Cloud Pak for AIOps, see Storage requirements.

A starter deployment requires 480 Gi of persistent storage, and a production deployment requires 1825 Gi of persistent storage. Expand the following sections to review details about storage requirements.

Persistent storage sizing

▶ Expand this section for details.

Additional storage requirements for offline (airgap) deployments

▶ Expand this section for details.

Additional storage requirements for backup and restore

▶ Expand this section for details.

IOPS requirements

The following table identifies the input/output operations per second (IOPS) per node that must be supported for a 3-node storage solution. Ensure that your hardware can support the expected IOPS per node. A minimum configuration of three nodes for the storage cluster is needed. Each node of the storage solution requires a minimum of one disk (SSD or high-performance storage array).

Table. IOPS requirements

Deployment size	Minimum	Recommended
Starter deployment	2,000	5,000
Production deployment	10,000	20,000

The results for your storage can vary depending on your exact usage, data sets, hardware, storage solution, and more. The preceding IOPS minimum and recommended values are based on a Red Hat® OpenShift® Data Foundation storage solution.

Processing abilities

Expand the following sections to find out about the processing abilities of starter and production sized deployments of IBM Cloud Pak for AIOps.

For more information about customizing the size of your IBM Cloud Pak for AIOps deployment according to your processing and footprint requirements, see Custom sizing.

Supported resource number and throughput rates for starter and production deployments

▶ Expand this section for details.

Event, alert, and incident rates

▶ Expand this section for details.

Custom sizing

The default starter and production deployment sizes enable the full capabilities of IBM Cloud Pak for AIOps to be used with the workload volumes that are stated in the Processing abilities section. If different workload volumes are required or resource constraints are an issue, then specific capabilities such as **`Metric Anomaly Detection`**, **`Log Anomaly Detection`**, and **`Runbook Automation`** can be sized accordingly. IBM Sales representatives and Business Partners have access to a custom sizing tool that can assess your runtime requirements and provide a custom profile that scales IBM Cloud Pak for AIOps components. The custom profile is applied when you install IBM Cloud Pak for AIOps.

Expand the following sections to view the processing abilities of some example custom-sized deployments of IBM Cloud Pak for AIOps.

Example 1 represents a minimally sized deployment of IBM Cloud Pak for AIOps for the evaluation of event management. It demonstrates event analytics, noise reduction, and the automation of issue resolution on a small topology. Metric and log anomaly detection and change risk assessment are de-emphasized. Example 1 requires 49 vCPU and 143 GB memory.

Example 2 represents a production deployment of IBM Cloud Pak for AIOps which is focused on event management capabilities. It supports event analytics, noise reduction, and the automation of issue resolution on a large topology. Metric and log anomaly detection and change risk assessment are de-emphasized. Example 2 requires 176 vCPU and 366 GB memory.

Example 3 shows the upper limits that IBM Cloud Pak for AIOps can be scaled to, across all of its capabilities.

Example 1

▶ Expand this section for details.

Example 2

▶ Expand this section for details.

Example 3

▶ Expand this section for details.

Adjusting PVCs

The PVC sizes in Persistent storage sizing provide adequate space for a user to begin working with IBM Cloud Pak for AIOps. Based on your actual usage and needs, the PVCs might need to be adjusted up to handle the workload when IBM Cloud Pak for AIOps is installed and running. In particular, CouchDB, Elasticsearch, and Kafka might need significantly more disk space. Expand the following sections to review details about custom sizings for these PVCs.

Cassandra

▶ Expand this section for details.

CouchDB

▶ Expand this section for details.

Elasticsearch

▶ Expand this section for details.

Flink

▶ Expand this section for details.

Kafka

▶ Expand this section for details.

Postgres

▶ Expand this section for details.

Requests and limits

In Kubernetes, requests define the resources that must be provided, and limits define the upper boundary of extra resources that can be allocated. Resource quotas can optionally be used to dictate the allowed totals for all requests and limits in a namespace. If a resource quota is set, then it must be at an adequate level to ensure that workload placement is not inhibited. If you do not deploy resource quotas, then skip this section.

The following table shows the requests and limits that the IBM Cloud Pak for AIOps installation namespace and the **`ibm-common-services`** namespace require for a typical starter or production deployment.

Table. Requests and limits requirements for the IBM Cloud Pak for AIOps installation namespace and the ibm-common-services namespace.

Resource name	IBM Cloud Pak for AIOps installation namespace (Starter)	IBM Cloud Pak for AIOps installation namespace (Production)	ibm-common-services namespace (Starter)	ibm-common-services namespace (Production)
requests.cpu	74	162	6	14
requests.memory (Gi)	188	372	10	30
requests.ephemeral-storage (Gi)	100	150	10	10
limits.cpu	210	360	26	65
limits.memory (Gi)	250	480	24	60
limits.ephemeral-storage (Gi)	325	450	20	20

Important: To allow sufficient resources for upgrading your deployment, apply a minimum 20% increase in resource quota requests or limits. Deployment of additional services impacts these values. Review these values regularly to ensure that they still meet your requirements.

Use the Red Hat OpenShift command line to create or edit a **`ResourceQuota`** object with these values. For more information, see Quotas in the Red Hat OpenShift documentation.

The following examples show the YAML to create **`ResourceQuota`** objects for the IBM Cloud Pak for AIOps installation namespace and the **`ibm-common-services`** namespace in a production deployment of IBM Cloud Pak for AIOps.

Example YAML file

▶ Expand this section for details.

Infrastructure Automation

Expand the following sections to review details about the minimum resource requirements for installing Infrastructure Automation. Use this information for sizing your environment when you are deploying Infrastructure Automation.

Infrastructure Automation can be installed with IBM Cloud Pak for AIOps or stand-alone (Infrastructure Automation only). Multi-region and multi-zone clusters are not supported.

Tip: Work with your IBM Sales representative to ensure that the allocated hardware resources for your environment are sufficient to best fit your business needs. For requirements for larger production deployments and to make sure that you meet the right level of resiliency, an IBM representative can work with you on your deployment plans. For Infrastructure Automation, you need to provide the following information:

- The number of cloud environments that you connect to.
- The number of OpenStack environments that you manage.
- The average number of VMs in each of the cloud environments.
- The number of on-premises virtualized environments that you manage. Identify whether it is VMware or KVM, and the number of VMs in each virtualized environment.

Resource allocation for the Red Hat OpenShift Container Platform control plane master nodes

▶ Expand this section for details.

Resource allocation for the Red Hat OpenShift Container Platform worker nodes

▶ Expand this section for details.

Additional requirements for offline (airgap) deployments

▶ Expand this section for details.

IBM Cloud Pak for AIOps ports

IBM Cloud Pak for AIOps uses ports to establish communication within Red Hat OpenShift Container Platform and to the external integrations. Review the list of ports that need to be exposed for use in inbound and outbound communication with IBM Cloud Pak for AIOps.

Inbound communication ports

Inbound communication ports are used to establish connection from external sources to IBM Cloud Pak for AIOps capabilities. IBM Cloud Pak for AIOps exposes the following ports to achieve this communication.

- *TCP 80*: Default HTTP port.
- *TCP 443*: Default HTTPS port.
- *TCP 4100*: Used for Tabular Data Stream to connect to the on-prem IBM Tivoli Netcool/OMNIbus.

Notes:

- TCP 80 and TCP 443 ports must be available on all nodes, including master, worker, and storage nodes. These are the standard ports for HTTP/HTTPS inbound connections as defined by the Red Hat OpenShift Container Platform routes. Red Hat OpenShift Container Platform routes provide an abstraction for an external access so you do not have to deal directly with load balancers, inbound controllers, and other networking components.
- Cloud Pak for AIOps pods use only the standard ports such as port 80 or port 443 as defined by Red Hat OpenShift Container Platform routes for HTTP/HTTPS inbound connections. The inbound connections from external sources can include browsers for accessing the console, an external Kafka client that needs to place data on a Kafka topic, Slack inbound data, and more.
- TCP 4100, TCP 16311, and TCP 9081 ports must be available on the worker nodes.

Connect to externally hosted probes

If you plan to connect to externally hosted probes to ObjectServers in Cloud Pak for AIOps, NodePorts are required. You can map any available port from the range of permissible ports for NodePort services to the target 4100 port. Hence, the port that matters from an inbound network perspective is the port that is used in the NodePort definition.

For more information about external probes that connect to ObjectServers (primary and backup), see Configuring the Cloud system.

Outbound communication ports

Cloud Pak for AIOps needs access to outbound ports to establish connection for probes and integrations.

- **IBM Tivoli Netcool/OMNIbus integration**

  For Netcool/OMNIbus integration, the bulk of data transfer is usually done over Tabular Data Stream port 4100 on the Netcool side. Cloud Pak for AIOps is able to establish a connection to 4100 port from each of the worker nodes. The Netcool/OMNIbus integration uses the IDUC port. IDUC indicates which alerts are inserted, deleted, or updated since the last IDUC cycle, which allows the integration to know what to request. This integration initially connects to ObjectServer by using Tabular Data Stream and then ObjectServer announces the host and port to use for IDUC. Therefore, firewall rules allow connectivity to the IDUC port.

  Notes:

  - If a firewall prevents Cloud Pak for AIOps connecting to Netcool on your VMs, you need to update the firewall rules to correct this situation.
  - The Netcool admin can modify the default 4100 port. It is recommended to engage the Netcool admin so that the admin can confirm all the ports for the Netcool components, such as ObjectServers, and Netcool/Impact.

  For more information about Netcool/OMNIbus to Cloud Pak for AIOps integration, see Creating IBM Netcool Operations Insight ObjectServer integrations.

  For more information about IDUC port, see the following links:

  - Set up the Netcool OMNIbus ObjectServer IDUC port.
  - Specifying the IDUC port
- **IBM Tivoli Netcool/Impact integration**

  The default ports for Netcool/Impact integration are 16311 and 9081. The 16311 port is used to access the GUI and the 9081 port is used to access the Netcool/Impact API.

  Netcool/Impact admin can decide to use different ports. It is recommended to get all the ports directly from admin in case the default ports have changed.

  For more information about Netcool/Impact to Cloud Pak for AIOps integration, see Creating IBM Tivoli Netcool/Impact integrations.

- **Probes**

  For probes that send events directly to Cloud Pak for AIOps, the IDUC port that is specified for the on-prem ObjectServer configuration needs to allow connectivity for Cloud Pak for AIOps. This IDUC port needs to be accessible from each of the worker nodes.

  In environments where a firewall is enabled between the servers or where iptables are configured on the servers, the `ncp_g_event` process can connect to the main ObjectServer listening port but might fail to connect to the IDUC port. While the main ObjectServer listening port uses a fixed port, the IDUC listening port is randomly selected by default. A fixed IDUC listening port must be specified to define firewall rules or iptables settings that allow IBM Tivoli Network Manager to connect to the ObjectServer on the IDUC listening port.

  For more information about IDUC port, see the following links:

  - Set up the Netcool OMNIbus ObjectServer IDUC port.
  - Specifying the IDUC port
- **Other integrations**

  For integrations such as Netcool, ELK, Splunk, Humio, LogDNA and topology observers, Cloud Pak for AIOps uses a PULL model.

The configurations of these integrations (endpoint exposed by the external source) dictate the port and connection parameters that Cloud Pak for AIOps uses to pull data, such as event and log from those external systems. The port number that is used by that external system depends on how the external integration was configured. Cloud Pak for AIOps does not dictate the value of port number. The outbound ports for each of these integrations need to be open on each of the worker nodes.

In some cases, these integrations can exist within the same intra-network where Cloud Pak for AIOps is installed. In other cases, you might be using a SaaS offering of the integrations (such as Instana or New Relic). For such cases, if you do not want to allow direct outbound access from the Red Hat OpenShift Container Platform cluster to the SaaS service, you can use the Secure Tunnel approach. This approach requires a VM with internet access, which allows a VM to connect to the SaaS instance.

For more information about the Secure Tunnel approach, see Secure Tunnel.

For more information about other integrations, see the following topics.

  - Dynatrace Events integration
  - Dynatrace Topology integration
  - Splunk Events integration
  - SevOne Events integration
  - SevOne Topology integration
  - ITNM Topology integration
- **Internal communication within the Red Hat OpenShift Container Platform cluster where Cloud Pak for AIOps is installed**

  Cloud Pak for AIOps internally uses other ports such as 8080, 8383 and 8686 for communication between its own pods, which run on the same Red Hat OpenShift Container Platform cluster.

Operator installation mode

Learn about the Red Hat® OpenShift® Container Platform operator installation modes that are supported by the IBM Cloud Pak® for AIOps operators.

For more information on Red Hat OpenShift operator installation modes, see Operator groups in the Red Hat OpenShift Container Platform documentation.

IBM Cloud Pak for AIOps

The IBM Cloud Pak for AIOps operator supports the Red Hat OpenShift **`OwnNamespace`** and **`AllNamespaces`** installation modes.

The **`All namespaces on the cluster (default)`** installation mode in the UI installation of the IBM Cloud Pak for AIOps operator is equivalent to the **`AllNamespaces`** mode in the CLI installation. In this mode, the IBM Cloud Pak for AIOps operator is installed in the **`openshift-operators`** project (namespace), and watches and is available to all namespaces in the cluster.

In the **`OwnNamespace`** installation mode, you must create and select a project (namespace) for the IBM Cloud Pak for AIOps operator to be installed into. In this mode, the IBM Cloud Pak for AIOps operator watches and is available for use only in this single namespace.

The **`OwnNamespace`** installation mode is the recommended installation mode for IBM Cloud Pak for AIOps, to provide better resilience and performance.

Suboperator installation mode

Suboperators are installed by the Operator Deployment Lifecycle Manager (ODLM) in the project (namespace) where the user created the **`Installation CR`** instance. Each suboperator watches and is available for use in its project (namespace).

For more information about configuring operator installation mode, see Adding Operators to a cluster.

IBM Cloud Pak for AIOps Infrastructure Automation

The Infrastructure Automation operator supports the OpenShift **`OwnNamespace`** and **`AllNamespaces`** installation modes when installed with, or without, IBM Cloud Pak for AIOps.

If you are installing with IBM Cloud Pak for AIOps with the **`OwnNamespace`** installation mode, you must follow the same instructions for IBM Cloud Pak for AIOps to use that installation mode.

High Availability (HA) considerations

You can set up high availability (HA) for your IBM Cloud Pak® for AIOps deployment by selecting a production (large) sizing profile.

IBM Cloud Pak for AIOps

HA configures your deployment with extra replication of its services, and affinity rules to provide redundancy. A HA configuration allows your deployment to tolerate upgrades and service interruptions to a portion of the infrastructure, such as a single node failure.

Sizing is configured in your IBM Cloud Pak for AIOps installation custom resource with the parameter **Capacity** (Red Hat® OpenShift® Container Platform UI) or **spec.size** (Red Hat OpenShift CLI), and can be set to **`small`** for starter deployments or **`large`** for production deployments. For a high availability deployment, you must set a value of **`large`** in your IBM Cloud Pak for AIOps **`Installation`** custom resource when you create your IBM Cloud Pak for AIOps instance.

For more information about sizing, see Hardware requirements.

A technology preview of installing IBM Cloud Pak for AIOps on a multi-zone architecture is also available. For more information, see Installing IBM Cloud Pak for AIOps on a multi-zone architecture (multi-zone HADR).

Infrastructure Automation

If you are installing Infrastructure Automation for use with IBM Cloud Pak for AIOps, you can help set up high availability for your IBM Cloud Pak for AIOps deployment by selecting a production (large) sizing profile.

If you are installing Infrastructure Automation as a stand-alone deployment and you are using Infrastructure management appliances, you can configure different settings to help you set up high availability. For more information, see High Availability Guide

Installing IBM Cloud Pak for AIOps on a multi-zone architecture (multi-zone HADR)

BETA The use of the *multi-zone HADR deployment architecture for IBM Cloud Pak for AIOps* is available only for technology preview, and is not for production usage.

Overview

IBM Cloud Pak® for AIOps is starting to implement multi-zone availability, as provided by the Red Hat® OpenShift® Container Platform.

Installing IBM Cloud Pak for AIOps on a Red Hat OpenShift cluster that spans multiple availability zones improves resiliency by enabling the distribution of IBM Cloud Pak for AIOps's services across nodes in different zones in the cluster. If a zone

becomes unavailable, then the remaining services in the other zones are still able to service requests, and disruption is lessened.

Failures in the following domains are minimized:

- individual pod failure: Red Hat OpenShift recovers individual failed pods.
- cluster node failure (master or worker): Red Hat OpenShift recovers non-stateful deployments.
- zone failure: Red Hat OpenShift reschedules the failed nodes' workloads to another zone.

For more information about the benefits of multi-zone architectures, see Running in multiple zones in the Kubernetes documentation.

Red Hat OpenShift tolerates nodes (and by extension zones) being down for five minutes, in addition to a node monitoring interval. For that reason, and accounting for the additional image pull and pod startup times, node and zone failures take roughly six minutes to start recovering. For more information, see Using remote worker node at the network edge: Kubernetes zones in the Red Hat OpenShift documentation.

Note: Multi-zone availability is only supported for production deployments of IBM Cloud Pak for AIOps. A starter deployment of IBM Cloud Pak for AIOps can be deployed on a multi-zone architecture, but it will not be highly available.

Considerations

Note the following for singleton operators, connections, and secure tunnel usage in a HADR environment.

Singleton operators

Most of the IBM Cloud Pak for AIOps operators are singletons (have only one instance). If a singleton operator is running in a zone that is subject to an outage, then the custom resource that the operator manages is not reconciled until Red Hat OpenShift detects the outage and reschedules the pod to an unaffected node or zone. Running functionality is not affected, but changes made to the configuration settings just before or during the outage are delayed until affected pods are rescheduled and running.

Connections

The following connections temporarily cease to function if they are running in a zone that is subject to an outage, but are automatically relocated to nodes in an unaffected zone by Red Hat OpenShift:

- Ansible
- Custom
- ELK
- Falcon LogScale
- Generic webhook
- IBM Tivoli Netcool
- Instana
- Kafka
- LogDNA
- Microsoft® Teams
- PagerDuty
- ServiceNow
- Slack
- SSH

Note: Other connections may cease to function until the zone experiencing the outage is restored, or a manual intervention is made.

Real-time data and actions are lost until the connection is functioning again. Any historical data that is being collected is delayed, not lost.

Secure tunnel

The Secure Tunnel function is enhanced to better handle zone outages. When you configure connector and tunnel worker replicas, ensure that the replica count is equal to or greater than the number of zones in your environment. The number of replicas is preferably a multiple of the number of zones. For example, if your environment has three zones then you might

configure two replicas for each zone, to give a total of six replicas. For more information about configuring connector and tunnel replicas, see Creating Secure Tunnel connections.

If one of the connection worker pods is not available for Secure Tunnel to use then performance might be impacted, depending on the current workload.

If the Secure Tunnel controller pod is running on a node or zone that is subject to an outage, then Red Hat OpenShift reschedules the pod to an unaffected node or zone. Until then, any changes to the tunnel connections or application mappings fail.

Known limitations and exclusions

Multi-zone HADR is not yet fully implemented for observers and probes. If the user interface or AI modelling are affected by an outage, then they are unavailable until Red Hat OpenShift relocates the corresponding pods.

Installing IBM Cloud Pak for AIOps on a multi-zone architecture

1. Prepare your cluster

   Before you install IBM Cloud Pak for AIOps, you must ensure that your Red Hat OpenShift cluster is configured for multi-zone by labeling the cluster nodes with a zone. Cluster administrator permission is required to add labels to your nodes.

   If you are installing IBM Cloud Pak for AIOps on a cloud platform, then refer to your cloud provider's documentation to learn how to select a multi-zone type when you are setting up your cluster, and then skip to step 2, Configure storage.

   If you are installing directly on Red Hat OpenShift on-premises, then label your Red Hat OpenShift cluster nodes as follows.

   1. Label cluster nodes with zones.

      Distribute your cluster nodes across three availability zones, ensuring that you have a cluster master node in each availability zone. Allocate each node to an availability zone by labeling the node with the zone:

      ```
      oc label node <node-name> topology.kubernetes.io/zone=<zone-name>
      ```

      Where

      - **<node-name>** is the fully qualified domain name of your node, for example *worker0.mycluster.subdomain.example.com*.
      - **<zone-name>** is the name of the zone that you want the node to be in.

   2. Label cluster nodes with regions.

      IBM Cloud Pak for AIOps has a dependency that requires the Kubernetes region label to be set on each node. The region label can be set to the same value on all nodes.

      ```
      oc label node <node-name> topology.kubernetes.io/region=<region-name>
      ```

      Where

      - **<node-name>** is the fully qualified domain name of your node, for example *worker0.mycluster.subdomain.example.com*.
      - **<region-name>** is the region name.

   When the **topology.kubernetes.io/zone** label is specified for the nodes in a cluster, Kubernetes tries to schedule a ReplicaSet's pods on different worker nodes in different zones. For example, a deployment of three replicas should result in one replica being deployed in each zone of a three zone cluster.

2. Configure storage

   Refer to your storage provider's documentation for instructions on configuring storage classes for a multi-zone configuration that enables data to remain available if a cluster node becomes unavailable.

3. Install IBM Cloud Pak for AIOps

   Starter deployments are not configured to support multi-zone HADR. Use the following instructions to install a production installation of IBM Cloud Pak for AIOps.

Installing IBM Cloud Pak for AIOps (Production installation)

Some resources might not failover as expected, and might require intervention. Contact IBM Support if you encounter any problems when you are using this technology preview.

Security considerations

IBM Cloud Pak® for AIOps supports different mechanisms for securing your environment and your data.

IBM Cloud Pak for AIOps builds on the security features provided by Red Hat OpenShift Container Platform by creating Security Context Constraints (SCC), service accounts, and roles. These features ensure that IBM Cloud Pak for AIOps pods and users have the lowest level of privileges to the Red Hat OpenShift Container Platform that is needed for them. These features protect sensitive customer data with strong encryption controls and improve the oversight of access control across applications and the platform itself.

In addition, ensure that you keep up to date on any security-related news or vulnerabilities with IBM Cloud Pak for AIOps by subscribing to security bulletins.

Security context constraints (SCC)

Administrators can use SCCs to control permissions for pods on their Red Hat OpenShift Container Platform cluster. These permissions include actions that a pod can perform and what resources it can access. Red Hat OpenShift Container Platform clusters contain default security context constraints (SCCs). IBM Cloud Pak for AIOps runs with the **`restricted`** or **`restricted-v2`** SCC that is included within this default set of constraints. For more information, see Red Hat® OpenShift® Container Platform SCCs and Red Hat - Managing Security Context Constraints.

Note: SecurityContextConstraints do not apply to the **`default`** or **`kube-system`** namespaces.

Permissions

For more information about the permissions that are required, see Permissions.

Volume level encryption

You can configure Portworx encryption or host-level encryption, depending on your requirements. Decide which level of encryption is needed for each stateful data store in your environment.

- For more information about using encrypted volumes on Red Hat OpenShift Container Storage, see Creating a storage class for persistent volume encryption.
- For more information about using the encryption feature for Portworx volumes, see Create encrypted PVCs.

Security hardening

For production environments, ensure that the Red Hat OpenShift Container Platform cluster where you are installing IBM Cloud Pak for AIOps is configured securely. To help harden the security for your cluster, consider completing the following tasks:

- Configure the Red Hat OpenShift Container Platform to enable etcd data encryption. For more information, see the Red Hat documentation about Encrypting etcd data.
- Implement a network egress policy that strictly controls what outbound calls can be made from applications in the cluster and from the IBM Cloud Pak for AIOps project (namespace). For more information, see the Red Hat documentation about Creating a network policy.
- If you use your own certificate authority (CA), replace the default certificate manager CA with your own CA.
- Ensure that your Red Hat OpenShift Container Platform and Kubernetes environment is configured according to the Center for Internet Security (CIS) Benchmark for Red Hat Enterprise Linux based OpenShift V4 and Kubernetes V1.6

security hardening. For more information, see the Center for Internet Security (CIS) Benchmarks about Securing Kubernetes.

- Configure lockout mechanisms on your authentication server to handle excessive login attempts.
- Shorten session timeouts to the shortest tolerable duration.
- Use your own certificate authority and monitor certificate expiration dates.
- Configure rate limiting connections for basic protection against distributed denial-of-service (DDoS) attacks. For more information about how to configure this limiting, see the Red Hat documentation about Route configuration.
- Limit users from viewing information about other users. By default, user IDs, user groups, roles, and permissions for other users are visible to users. To block non-administrative users from viewing user groups, roles, and permissions,

  1. Edit the config map **`product-configmap`** to add the configuration **`usermgmt_limit_user_info: "true"`**.
  2. Restart the user management pods, by running the following command:

     ```
     oc delete pod -l component=usermgmt
     ```

- If your installation does not use a load balancer or proxy that depends on http "HOST" headers, enable host header injection protection:

  1. Edit the config map **`product-configmap`** to set the following values:
     - Ensure that the host headers are being used to match the value of **`URL_PREFIX`** from the **`product-configmap`**.
     - Set the **`HOST_INJECTION_CHECK_ENABLED`** value to **`true`** in the **`product-configmap`**.
  2. Restart the **`nginx`** pods.

Certificates

If you want to use a custom certificate for IBM Cloud Pak for AIOps, or Infrastructure Automation, then you can do this with either of the following methods:

- Configure a custom certificate for the Red Hat OpenShift cluster. Follow the instructions in the Red Hat OpenShift documentation Replacing the default ingress certificate. Then, deploy the signing CA certificate into the cluster by following the instructions in the Red Hat OpenShift documentation Replacing the CA Bundle certificate.
- If you would like to use a custom certificate for IBM Cloud Pak for AIOps or Infrastructure Automation only, then after installation is complete follow the instructions in Using a custom certificate.

Image verification

You can optionally verify the IBM Cloud Pak for AIOps images before you install IBM Cloud Pak for AIOps. For more information, see Verifying images.

Verifying images

Use this information to verify the signatures on the IBM Cloud Pak® for AIOps images.

Digital signatures provide a way for consumers of content to ensure that what they download is authentic (it originated from the expected source) and has integrity (it is what it is expected to be). All images for IBM Cloud Pak for AIOps are signed.

Prerequisites

- Ensure that you these command line tools are installed (they can usually be installed on Linux® with the package manager):
  - GNU Privacy Guard v2 (package name **`gpg2`**)

    - OpenSSL (package name `openssl`)
    - skopeo
- The IBM Cloud Pak for AIOps public key must exist on the same machine as the command line tools. Copy the following text block exactly as shown into a text editor, and save it in a file named **cp4waiops-public.pub.asc**:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGPJiXwBEADTRz2VX7ukVRhPvwpq+LUhbc9pwCIXCA2Mj9M5/OIY7UEZ57+9
bs0pKbumxa4H6JOKl4iQd+jdlmeZU5W6jeJoDqQ7YPIniKemnqtbGqIDNN7bvkAe
t7hcX4QdgQTezO2r7AsEkjRnTioXBhP1Ie+RHbnZlWRexwSa2GPaZmewpH6ysFBI
6mysLpXiBRs0vRtNwPvliBjZIwtLXZYPF83mt2YvB2uzrRrDPD9OCfNW8vUpdvuu
EsGHps/LzV4iob6TWWcbQSvagnq3KG8/Z6CAEOWDCJFR78kgqtm0bN/SKXFxFXMi
n+OLwNhLm4E+l0aJ3lGUCfjAhY5q4WftUPoytnNLrolyKoHSCM08h1YGm0EB/xVQ
z6+sl+YmXtV0QEx+yyaV51IONXKHBsAWiyoiOr7KmHF6SPIbxXPK13QQqgkeqRAU
MY1t9lUxSuVugdImBeVtNBf5iSJp7FYMAzF4q8Tpr2KzB3gPyM+f9jf89+ZpiUZg
LsX7Xp3Kl6TlwvIU+HU2LHhBJZ6KnQgUou7Q7y4gIeypnUDmb3lZwg5eRQBvA4sQ
xS2Ct3+nBM9g991FnFzK0N7YrdjgyiODx0QLY7RG+T2MHwaQj0apjy23qY5Otnix
3YtDK4qwazp14ciaQY02eFtQxtk0Vwfom83n/QTO8QCs6aBGA8RcpaBFuwARAQAB
tDFJQk0gQ2xvdWQgUGFrIGZvciBXYXRzb24gQUlPcHMgPHBzaXJ0QHVzLmlibS5j
b20+iQI6BBMBCAAkBQJjyYl8AhsPBQsJCAcCBhUKCQgLAgQWAgMBAh4BBQkDwmcA
AAoJEBjaoXhlVSaRzZEQAL978PoyTAhusQEHHgOMo/cjMFLtCnP9YsPHWjK6Q4Gf
rRE8LUNS79IMwRe4vRFaNiDmjB3iTSaFpcwSYeRJJi+V5a2a7FuKRqEUx81AI3qy
r8/V5tPlKD0uTh+vXGb0/Suot5h0A1ft9nrMfilt84JbQRGTiVi0obcmhJPM7H07
B4j7453BsEzMaDOYNpqvgzt+wu/fWeJg6ALpEY4Tl4qwHn4gQCNB3MYM1DFmqgZr
CnRr0iKTI7gfpRIbxdBqXowWDs9KRCIVXcjXEpMYz6I9XM5mN+OEgc/7VIW0eXbB
nEUPozm8VRDJRGh8vc9FMES6kLO28hT7YM6pwv91vkPQD4T/dhAVoCz30rgVBvBJ
ftYR7eWKBNe97cXJnMmg4kBxO0UiCW6mhtrgdgDdGtT6nvbCof5n9IeaRRxM5DOW
HM4JKIjk5cSJOHpgRqjvCTc8bpjWQruJ6lcm+KR/D5ewNOe14dHs6ssBWJUZqdnk
tFg/ihQXBv+y77hR+JXPCmpgpZmBKeJGqDFWvfO6fsQw06zlz2cJI2tQ+xApHmt3
dxUKhow+trihE1ZAVPXsXe4U3vVt97jR4AUYQxDlszbvuTOsO3jdRhMhy5/RyH4w
rsGfS6ies5LZqdGSszt0E09jAC9jYHb3brdOyScPkkTNr8DPyD/u/oDH21bFNCzk
=2uZA
-----END PGP PUBLIC KEY BLOCK-----
```

- You must have a list of images to verify. To get a list of container images used in Cloud Pak for AIOps, refer to the procedure in Downloading or listing container images. In the following procedure, the example image *cp.icr.io/cp/cp4waiops/node-server:v3.7.0-00000000.0000-000000000* is used.

  Note: This tag is an example for demonstration purposes; it is not a real tag. Obtain the most up-to-date image names and tags with the procedure in Downloading or listing container images.

Procedure

1. Log in to the entitled registry **cp.icr.io** so that you can pull images from that registry.

   ```
   podman login cp.icr.io --username cp --password <entitlement_key>
   ```

   If you need to obtain the entitlement key that is assigned to your ID, complete the following steps:

   1. Log in to MyIBM Container Software Library with the IBMid and password that are associated with the entitled software.
   2. In the Entitlement keys section, select **Copy key** to copy the entitlement key to the clipboard.
2. Import the IBM Cloud Pak for AIOps public key on the machine that you prepared according to the Prerequisites section:

   ```
   sudo gpg2 --import cp4waiops-public.pub.asc
   ```

   Note: This step needs to be done only once on each machine you use for signature verification.

3. Calculate the fingerprint:

   ```
   fingerprint=$(sudo gpg2 --fingerprint --with-colons IBM Cloud Pak for AIOps | grep fpr |
   tr -d 'fpr:')
   ```

   This command stores the key's fingerprint in an environment variable called *fingerprint*, which is need for the command to verify the signature. When you exit your shell session, the variable is deleted. The next time that you log in to your machine, you can set it again by rerunning the command.

4. Create a directory for the image and use skopeo to pull it into local storage:

```
mkdir images
skopeo copy docker://cp.icr.io/cpopen/ibm-aiops-orchestrator:3.2.0-2021-00-00-0000-
00000000 dir:$HOME/images
```

This command downloads the image as a set of files and places them in the images directory (or another directory that you choose).

Note: There is a manifest file named *images/manifest.json*, and a signature file named *images/signature-1*. You reference both these files in the next step (in the command to verify the signature).

5. Verify the signature:

```
for SIGNATURE in $(ls $HOME/images | grep signature); do
    skopeo standalone-verify $HOME/images/manifest.json cp.icr.io/cp/cp4waiops/node-
server:v3.7.0-00000000.0000-000000000 ${fingerprint} $HOME/images/$SIGNATURE 2> /dev/null
    if [ $? -eq 0 ]; then
       break
    fi
done
```

If the signature is verified, you get a confirmation similar to the following output. Otherwise, no output is provided:

```
Signature verified, digest
sha256:0000000000000000000000000000000000000000000000000000000000000000
```

Downloading or listing IBM Cloud Pak for AIOps images

You might need to download all of the container images for IBM Cloud Pak for AIOps, for example, to copy them to an internal registry, scan them for vulnerabilities, or verify their signatures. This procedure describes how to download container images, although you may need to use different tools, depending on your particular environment.

1. Follow the procedure in Installing IBM Cloud Pak for AIOps in an air-gapped environment (offline) using a bastion host, up to and including the "Section 2 Download the CASE".

   These steps:

   - download the product CASE packages, which contain metadata for the container images required to deploy each capability. Each CASE package also contains the required scripts to mirror images to a private registry, and to configure the target cluster to use the private registry as a mirror.
   - generate comma-separated value (CSV) files listing the images for all capabilities (one CSV file is created for each capability and required dependencies). The CSV files, combined with your IBM Entitled Registry entitlement key, are used to download or mirror the images manually in the scripts below.

2. Log in to your container tool to access the entitled registry, using the username **cp** and your entitlement key.

3. Use a shell script to process all images from all capabilities supported on the target architecture.

   For example, for x86/amd64 architecture:

```
tail -q -n +2 $HOME/offline/*-images.csv | while IFS="," read registry image_name tag
digest mtype os arch variant insecure digest_source image_type groups; do
if [[ "$arch" == "amd64" ]]; then
  docker pull "$registry/$image_name@$digest"
  docker tag "$registry/$image_name@$digest" "$registry/$image_name:$tag"
fi
done
```

   Instead of docker, you can use alternative container tools such as podman or skopeo.

   - Use podman as a direct replacement for docker in the preceding commands.
   - Use one skopeocommand to replace the docker pull and docker tag commands. For example:

```
skopeo copy --dest-tls-verify=false docker://$registry/$image_name@$digest docker-daemon:$registry/$image_name:$tag
```

You can also use skopeo to save images to disk, or to copy them directly to another container registry.

Note: If you need your clusters to access these images from a private registry, the image digests must be preserved. If this is your intention, IBM recommends using the procedure in Installing IBM Cloud Pak for AIOps in an air-gapped environment (offline).

Licensing

Learn more about the licensing and entitlements for IBM Cloud Pak® for AIOps.

License entitlements

A IBM Cloud Pak® for AIOps license provides entitlement to deploy:

- IBM Cloud Pak for AIOps 4.2.0
- IBM® Tivoli® Netcool®/OMNIbus
- IBM AIOps Metric Manager

In terms of capabilities with IBM Cloud Pak for AIOps there are three categories of licensed entitlement:

- Event processing and analytics
- Metric processing and analytics
- Log processing and analytics

Each entitlement provides access to different capabiliites. To access the entitled capabilities, you need to deploy the required configuration. For more information about the supported deployment scenarios, see Deployment scenarios.

The following table identifies the general capabilities available with each entitlement and the instance that must be deployed to access the capabilities.

License entitlements and installations

Entitlement	Description	Required installation for analysis and use
Event processing and analytics	- Event processing and management - Common capabilities	• IBM Cloud Pak for AIOps
Metric processing and analytics	- Metric and KPI processing, analytics, anomaly detection, and forecasting - Common capabilities	- IBM Cloud Pak for AIOps OR - Metric Manager (Predictive Insights)
Log processing and analytics	- Log processing, analytics, and anomaly detection - Common capabilities	• IBM Cloud Pak for AIOps

License tracking

As part of using IBM Cloud Pak for AIOps, tracking your license usage is required. Although IBM Cloud Pak for AIOps provides license options not tracked directly by the product, to track usage, you must deploy the IBM Cloud Pak foundational services License Service on all clusters where IBM Cloud Pak for AIOps is installed.

IBM Cloud Pak foundational services License Service

This background License Service collects and stores the license usage information at the cluster level for tracking license consumption and for audit purposes. You can retrieve this data upon request for monitoring and compliance. You can also retrieve an audit snapshot of the data that is audit evidence.

Note: For IBM Cloud Pak for AIOps, the license service collects usage data for only IBM Cloud Pak for AIOps. Although IBM Cloud Pak for AIOps provides license options not tracked directly by the product, to track usage, you must deploy the IBM Cloud Pak for AIOps License Service on all clusters where IBM Cloud Pak for AIOps is installed.

Learn about how License Service collects and measures the license usage of IBM Cloud Pak for AIOps, and how to retrieve the license usage data.

- Overview
- Hyperthreading
- Audit snapshot
- Reported metrics
- Enabling optional features
- Viewing license usage on the Administration panel
- Retrieving license usage data
- Verifying completeness of license usage data and troubleshooting
- Using custom certificates
- Tracking license usage in multicluster environment
- Tracking license usage with License Service Reporter
- Deploying License Service Reporter
- Configuring data sources
- Validating the deployment
- Viewing license usage on the Licensing dashboard
- Manually tracking license usage

License Service overview

License Service collects and measures the license usage of IBM Cloud Pak for AIOps at the cluster level. You can retrieve this data upon request for monitoring and compliance. You can also retrieve an audit snapshot of the data that is audit evidence.

Note: For IBM Cloud Pak for AIOps, the license service collects usage data for only IBM Cloud Pak for AIOps. Although IBM Cloud Pak for AIOps provides license options not tracked directly by the product, to track usage, you must deploy the IBM Cloud Pak for AIOps License Service on all clusters where IBM Cloud Pak for AIOps is installed.

About License Service

License Service is required for monitoring and measuring license usage of the IBM Cloud Paks® and IBM® stand-alone containerized software in accord with the pricing rule for containerized environments. Manual license measurements are not allowed.

License Service

- Collects and measures the license usage of Virtual Processor Core (VPC) metric at the cluster level.
- Collects and measures the license usage of IBM Cloud Paks and their bundled products that are enabled for reporting and licensed with the Managed Virtual Server (MVS) license metric.
- Collects and measures the license usage of Virtual Processor Core (VPC) and Processor Value Unit (PVU) metrics at the cluster level of IBM® stand-alone containerized software that is deployed on a cluster and is enabled for reporting. For more information, see License Service for stand-alone products.
- Currently, License Service refreshes the data every 5 minutes. However, this might be subject to change in the future. With this frequency, you can capture changes in a dynamic cloud environment.
- Provides the API that you can use to retrieve data that outlines the highest license usage on the cluster.
- Provides the API that you can use to retrieve an audit snapshot that lists the highest license usage values for the requested period for products that are deployed on a cluster.

Note: Only one instance of License Service is deployed per cluster regardless of the number of IBM Cloud Paks and containerized products that you have installed on this cluster.

Supported platforms

- License Service that is integrated into an IBM Cloud Pak as a part of IBM Cloud Pak® foundational services is supported on Linux® x86_64, Linux® on Power® (ppc64le), and Linux® on IBM Z® and LinuxONE with Red Hat® OpenShift® Container Platform version 4.3, 4.4 and 4.5.

  Note: To deploy License Service and collect data for IBM Cloud Paks that are deployed on older versions of Red Hat® OpenShift® Container Platform, such as 3.11 to 4.3, follow the deployment procedure for License Service for stand-alone products.

- You can deploy License Service on other Kubernetes-orchestrated clouds to collect license usage of IBM stand-alone containerized software. For deployment instructions and the full list of supporter platforms, see License Service for stand-alone products.

Hardware requirements

License Service uses the following hardware resources:

- **CPU limits**: 500 m
- **Memory limits**: 512 Mi

Deployment

License Service is automatically deployed as a part of IBM Cloud Pak for AIOps, and has minimal impact on cluster capacity. License Service works in the background and does not require any configuration or user engagement.

To make sure that this service is successfully deployed, see: Validating License Service deployment.

Storage

License Service does not require persistent storage.

Using License Service for container licensing

After the automatic deployment, License Service collects and measures the license usage of the products that are deployed on the cluster. This data is required for compliance and audit purposes. With License Service, you can use the dedicated APIs to retrieve an audit snapshot for the cluster.

Core license metrics

Currently, supported core-based metrics for container licensing are Processor Value Unit (PVU) and Virtual Processor Core (VPC). For core license metrics, you are obliged to use License Service and periodically generate an audit snapshot to fulfill container licensing requirements. License Service is an automatic solution. You do not need to complete any manual actions to prepare the audit snapshot, you only need to generate it. At this point, the audit snapshot is required to be generated at least once a quarter for the last 90 days, and stored for 2 years in a location from which it can be retrieved and delivered to auditors.

Note, that the requirements might change over time. Always make sure to follow the latest requirements that are posted on Passport Advantage.

For more information, see the following resources:

- IBM Container Licenses on Passport Advantage
- Container licensing FAQs
- Retrieving an audit snapshot

Non-core metrics

License Service additionally collects information about license usage of IBM® software that is enabled for reporting and licensed with other non-core metrics. You can use this data to track the license usage of these products, and determine your entitlement needs. For more information, see Reported metrics.

Best practices

- It is recommended to generate an audit snapshot report monthly as a precaution.
- Before decommissioning a cluster, record the license usage of the products that are deployed on this cluster by generating an audit snapshot until the day of decommissioning.
- Plan your storage to contain regular audit snapshots. The size of an audit snapshot `.zip` package might vary and depends on the number of products and the range of the reporting period. On average, the size of the package for a starter deployment environment is around 10 KB, and around 100 KB for production environments.

Tracking license usage in multicluster environment

Note: In multicluster environment, the cumulative report is not required for audit.

You can use the data that is collected by License Service from individual clusters to track the cumulative license usage in a multicluster environment. For detailed information, see Tracking license usage in a multicluster environment.

You can use License Service Reporter for that purpose. License Service Reporter is not obligatory. It is an extension of License Service that aggregates license usage data from multiple clusters and gives you clear view for the entire environment. License Service Reporter is deployed on one cluster and uses extra resources for aggregation. To use this functionality, extra configuration is needed. For more information, see Tracking license usage in multicluster environment with License Service Reporter.

If you prefer to track license usage without deploying License Service Reporter, you can follow the non-automated procedure. For more information, see Manually tracking license usage in multicluster environment.

Hyperthreading

License Service supports multiple threads per physical core also referred to as Simultaneous multithreading (SMT) or Hyper-Threading (HT).

Simultaneous multithreading (SMT) or Hyper-Threading (HT) is a technology that facilitates more efficient utilization of your processor's resources by enabling multiple threads to run on a physical core.

For more information, see Intel® Hyper-Threading Technology and Simultaneous Multithreading in Red Hat Enterprise Linux.

License Service supports worker nodes with Hyper-Threading or SMT variants SMT2, SMT4 and SMT8 enabled. For more information about the variants that are supported by IBM Container Licensing rules, see Additional License Counting Guidance for IBM Container licenses in Passport Advantage.

If your IBM software is deployed on a cluster that has SMT or HT enabled, it can significantly change your license costs for this clusters.

Example

The following example shows how Virtual Processor Cores are counted for the worker node with and without Hyper-Threading or SMT enabled.

Scenario: An IBM Cloud Pak® with two bundled products that are licensed with Virtual Processor Core (VPC) is deployed on a cluster that runs on Linux® x86_64. For simplicity, the assumption is that both bundled products are deployed on one worker node with the capacity of 8 Cores.

Table. Cores for worker node with Hyper-Threading enabled

HT/SMT enabled	Highest license use for bundled product 1	Highest license use for bundled product 2	Overall license use
No	CPU limit: 10 vCPU Node capacity: 8 vCPU Capped CPU limit: 8 vCPU Rounded up value: 8 VPC	CPU limit: 2.5 vCPU Node capacity: 8 vCPU Capped PCU limit: 2.5 vCPU Rounded up value: 3 VPC	**11 VPC**

HT/SMT enabled	Highest license use for bundled product 1	Highest license use for bundled product 2	Overall license use
Yes: HT	HT factor: 2 CPU limit: 10 vCPU Node capacity: 8 vCPU Capped CPU limit with HT factor: 4 vCPU Rounded up value: 4 VPC	HT factor: 2 CPU limit: 2.5 vCPU Node capacity: 8 vCPU Capped CPU limit with HT factor: 1.25 vCPU Rounded up value: 2 VPC	**6 VPC**
Yes : SMT4	SMT factor: 4 CPU limit: 10 vCPU Node capacity: 8 vCPU Capped CPU limit with SMT factor: 2 vCPU Rounded up value: 2 VPC	SMT factor: 4 CPU limit: 2.5 vCPU Node capacity: 8 vCPU Capped CPU limit with SMT factor: 0.625 vCPU Rounded up value: 1 VPC	**3 VPC**

Checking if the cluster had multiple threads per physical core is enabled

Before you configure License Service for Hyper-Threading or SMT, make sure that your cluster is enabled for that. Contact your infrastructure provider or read the relevant documentation to check the configuration on the infrastructure level.

Useful links:

- Red Hat Customer Portal: How can I tell if Hyperthreading (HT) is enabled on my system?
- Red Hat Customer Portal: What is hyperthreading and how do I configure it?
- Hyperthreading in VMware documentation
- Managing Hyper-V hypervisor scheduler types
- Disabling Intel Hyper-Threading Technology on Amazon Linux
- Azure compute unit (ACU)
- The tools for checking the status of SMT on IBM Power
- Simultaneous multithreading on IBM Z

Enabling hyperthreading in License Service

License Service supports hyperthreading on worker nodes. If your clusters have SMT or HT enabled, you need to enable hyperthreading in License Service. For instructions, see Enabling hyperthreading.

Hyperthreading in audit snapshot

Audit snapshot contains information about hyperthreading on a cluster. The information about the cluster threads per core is included in the **`data_condition.txt`** file.

Audit snapshot

Audit snapshot is a record of license usage in your environment over a period of time. The audit snapshot is a compressed **`.zip`** package that includes a complete set of audit documents that certify your cumulative license usage.

Audit snapshot for container licensing

Audit snapshot is needed for compliance and audit purposes.

For core license metrics, you are obliged to use License Service and periodically generate an audit snapshots to fulfill container licensing requirements. For more information about core license metrics, see Reported metrics.

You do not need to complete any manual actions to prepare the audit snapshot, you only need to generate it.

At this point, the audit snapshot is required to be generated at least once a quarter, and stored for 2 years in a location from which it could be retrieved and delivered to auditors.

Note: The requirements might change over time. You should always make sure to follow the latest requirements that are posted on Passport Advantage.

For more information, see the following resources:

- IBM Container Licenses on Passport Advantage
- Container licensing FAQs
- Retrieving an audit snapshot

Also, the Cloud pak for Data services usage information can be imported to the License Service Reporter in the following conditions:

- If you use audit snapshot upload API for offline scenarios. For more information, see Audit snapshot in a multicluster environment.
- You must have the License Service version 1.18.0 and License Service Reporter version 1.18.0.

Best practices

- It is recommended to generate an audit snapshot report monthly as a precaution.
- Before decommissioning a cluster, record the license usage of the products that are deployed on this cluster by generating an audit snapshot until the day of decommissioning.
- Plan your storage to contain regular audit snapshots. The size of an audit snapshot `.zip` package might vary and depends on the number of products and the range of the reporting period. On average, the size of the package for a small environment is around 10 KB, and for medium and large environments - around 100 KB.

Generating an audit snapshot

You can retrieve an audit snapshot for the cluster with the dedicated License Service API. For more information, see Retrieving an audit snapshot.

Content of the audit snapshot

The audit snapshot is a compressed `.zip` package that includes a complete set of audit documents that certify your cumulative license usage.

An audit snapshot might consists of the following files:

Table. Audit snapshot files

File name	Content
`checksum.txt`	The unique checksums that are a proof that the audit snapshot was not tampered with.
`data_condition.txt`	Audit snapshot metadata that includes the following information: • License Service or License Service Reporter version that was used to generate the snapshot • Information about cluster or clusters • Metrics that are included in the snapshot • Time and date when the report is generated • Reporting period: Start Date and End Date • Information about optional capabilities, such as, hyperthreading, or any additional Custom Resource definitions that are delivered by the product
`data_condition.json`	Audit snapshot metadata that includes all information that are listed in `data_condition.txt` and additionally information about custom cluster names and IDs that you defined in `.json` format. In the future, `data_condition.json` will replace the `data_condition.txt` file. To have all current information in `data_condition.json` in the snapshots generated in the License Service Reporter, you must use the License Service version 1.18.0 or newer, and License Service Reporter version 1.18.0 or newer.

File name	Content
`products_<reported_period>_<cluster hostname>.csv`	The aggregated highest license usage that is registered for each product within the reported period.
`products_daily_<reported_period>_<cluster hostname>.csv`	The aggregated highest license usage that is registered for each product within the reported period per day.
`bundled_products_<reported_period>_<cluster hostname>.csv`	The aggregated highest license usage for each bundled product that is a part of the IBM Cloud Paks.
`bundled_products_daily_<reported_period>_<cluster hostname>.csv`	The aggregated highest license usage that is registered for each bundled product within the reported period per day.
`pub_key.pem`	The public key file that can be used to verify the `signature.rsa` file against the `checksums.txt` file.
`signature.rsa`	A digital signature that can be used to verify whether the `checksums.txt` file was tampered with.
`unrecognized-apps-<reported_period>.csv`	A list of pods from which the license usage data was not collected on a specified date. The pods have incomplete or missing product annotations that provide the product metadata that is needed for measurements. The information is provided for every date within the reported period. The list contains the namespace followed by a pod name. You must have the License Service version 1.18.0 or newer, if you want to have the actual data in this file that is generated in License Service Reporter version 1.18.0 or newer.
`services_<reported_period>.csv`	The aggregated highest license usage for each Cloud Pak for Data services that are a part of bundled products.
`services_daily_<reported_period>.csv`	The aggregated highest license usage that is registered for each Cloud Pak for Data services within the reported period per day.

Understanding the audit snapshot

The list of columns in the audit snapshot `.csv` files.

Table. Audit snapshot columns

Column	Description
cloudpakId	The identification number of the IBM Cloud Pak® to which the program is bundled.
name	The name of the product.
cloudpakMetricName	The license metric unit that is used by the entire IBM Cloud Pak® to which the bundled product contributes.
cloudpakVersion	Version of the IBM Cloud Pak® to which the program is bundled.
clusterId	The identification of the cluster for which the highest license usage is calculated.
date	The date for which the metricQuantity or metricMeasuredQuantity is calculated.
id	The identifier of the product.
metricConversion	The ratio that shows how the license usage of the bundled product is counted when compared with the license usage of the IBM Cloud Pak®. It shows how the program's license metrics are recalculated when compared to the IBM Cloud Pak® license metrics.
metricConvertedQuantity	The number of license units that the bundled product contributed to the overall license usage of the IBM Cloud Pak®. The value is calculated by comparing metricMeasuredQuantity against metricConversion.
metricMeasuredQuantity	The highest number of license units that the bundled product used within the requested period.

Column	Description
metricName	The license metric unit that is used by the product.
metricPeakDate	The date when the license metric usage of the product was the highest within the requested period.
metricQuantity	The highest number of license units that the product used within the requested period.
productName	The name of the detected bundled product.
productId	The identifier of the bundled product.
serviceName	The name of the Cloud Pak for Data service that is a part of the bundled product.
serviceId	The identifier of the Cloud Pak for Data service that is a part of the bundled product.
serviceMetricValue	The license metric used by the Cloud Pak for Data service that is a part of the bundled product.

Audit snapshot in a multicluster environment

Note: License Service Reporter is only available with IBM Cloud Paks.

If you deploy and configure License Service Reporter, you can retrieve the audit snapshot for multiple clusters directly from the Licensing dashboard, or using the License Service Reporter API.

Audit snapshots that can be created in Kubernetes clusters and IBM License Metric Tool (ILMT):

- For all Kubernetes clusters: The Audit Snapshot that is created with action button of License Service Reporter, is the collection of Audit Snapshots identical in content. These Audit Snapshots are created on-demand in all Kubernetes clusters by License Services. Therefore, Audit Snapshot collection created in License Service Reporter can be used to facilitate preserving Snapshots from all Kubernetes.
- In IBM License Metric Tool (ILMT): As per IBM compliance, an audit snapshot is not valid even if the audit snapshot package information is uploaded from ILMT to License Service Reporter and that information is used to create a file with format similar to Audit Snapshots files for containerized environments. For non-containerized environments measured by ILMT, only Audit Snapshot created in ILMT is legitimate for Compliance purpose.

Audit Snapshot created by ILMT contains additional information that is specific for non-containerized deployments.

In offline scenario, to import the services usage data version 1.18.0 into the License Service Reporter version 1.18.0, you must use audit snapshot upload API that contains the services information, by using **`PUT`** request on **`/snapshot`** endpoint. The **`PUT`** request must contain the zip file that is generated from the License Service 1.18.0, that has Cloud pak for Data services. For more information, see Uploading audit snapshots from offline environments into License Service Reporter.

For more information about how to generate audit snapshot for multiple clusters, directly from the Licensing dashboard, see Generating audit snapshot from the Licensing dashboard. To learn how to retrieve the audit snapshot by using the dedicated API, see Retrieving an audit snapshot for multiple clusters.

Reported metrics

Learn more about the license metrics that are measured and collected by License Service.

Core metrics

License Service is required for monitoring and measuring license usage of the IBM Cloud Paks® and IBM® stand-alone containerized software in accord with the pricing rule for containerized environments. Manual license measurements are not allowed.

License Service collects the license metrics that are eligible for container licensing and allows you to generate the audit report for their usage at the cluster level. This data is required for compliance and audit purposes.

Table 1. License metrics eligible for container licensing

Metric	Description

Metric	Description
Virtual Processor Core (VPC)	A Virtual Processor Core is a license metric unit that is based on the number of virtual cores (vCPUs) that are available to the product. A vCPU is a virtual core that is assigned to a virtual machine or a physical processor core if the server is not partitioned for virtual machines.
Processor Value Unit	A Processor Value Unit (PVU) is a license metric unit that is based on the type and number of processors that are deployed on a server where the software is installed. For PVU-based licensing, IBM® defines a processor as a processor core on chip.

Currently, License Service refreshes the data every 5 minutes. However, this might be subject to change in the future. With this frequency, you can capture changes in a dynamic cloud environment.

Non-core metrics

License Service collects information about the license usage of IBM® products that are licensed with other non-core metrics. These metrics are not supported by container licensing. However, you can track their license usage and determine your entitlement needs.

Table 1. Other non-core metrics.

Metric	Description
Managed Virtual Server	A Managed Virtual Server (MVS) is a license metric unit that is based on the server instance whether it is physical or virtual. It eliminates the need to know what infrastructure the server runs on or what type of hardware resources the server being managed is consuming.

Information about license usage of the products that use non-core metrics is delivered to License Service regularly. In most cases, the licensing data is delivered daily. For some products, the data can be delivered even a few times a day. After each delivery, License Service recalculates all available data for the day, analyses all values, and then reports the highest license usage for the product.

Each data transfer can deliver information from the last seven days. Then, License Service recalculates the license usage of the product for this period, and shows the newly calculated highest usage. The recalculation does not impact the values prior to this seven-day period.

Note: Because the license usage data for non-core metrics can be delivered with different frequency and for up to seven days back, the values on the reports that you retrieve might differ.

Enabling optional features

You can enable the optional features of License Service to get extra benefits.

License Service works in the background and does not require any configuration for license compliance. However, you can perform optional configuration to enable additional License Service features and get extra benefits. By default, the features are disabled and transparent. To enable these features, edit the IBMLicensing custom resource.

Available features:

- Chargeback: Filtering the license usage data by user-defined groups - Use this feature to filter the license usage data by groups that are based on namespaces. Thanks to grouping, you can see how the internal organizations or business divisions in your company contribute to the overall license usage. This information might also help you monitor the namespaces during migrating your software to Cloud, or with establishing the potential chargeback.
- Enabling hyperthreading - Use this feature to enable License Service to properly count your license usage on the clusters where Simultaneous multithreading (SMT) or Hyper-Threading (HT) are enabled. When you enable hyperthreading, your license usage on clusters with SMT or HT are aggregated with regard to the hyperthreading rules.

Editing the IBMLicensing custom resource

Learn how to edit the **`IBMLicensing`** custom resource to enable the additional features.

- From the OpenShift console:

1. Log in to your OpenShift console.
2. Go to **Operators** > **Installed Operators**.
3. From the **Project** drop-down list, select **All Projects**
4. Select **IBM Licensing Operator**.
5. Select the **IBM License Service** tab. The instance of License Service custom resource is listed.
6. From the overflow action menu, select **Edit IBMLicensing**.
7. Edit the YAML. For example, to enable chargeback, add the following line: `chargebackEnabled: true` under `spec`.

   ```
   spec:
     datasource: datacollector
     chargebackEnabled: true
     httpsEnable: true
   ```

8. Click **Save**.

- From the command line:

  1. Run the following command:

     ```
     oc edit IBMLicensing instance -n ibm-common-services
     ```

     Note: The command assumes that the IBM Cloud Pak foundational services are installed in `ibm-common-services` namespace. If IBM Cloud Pak for AIOps installs foundational services in a different namespace, adjust the command.

  2. Make your changes to the YAML.

Chargeback - Filtering the license usage data by user-defined groups

This feature is available from IBM Cloud Pak® foundational services version 3.7.x.

For license compliance purposes, License Service automatically collects information about the highest license usage of your products in all namespaces in the entire cluster. However, for internal accounting, you might need to calculate the expenses associated with the license usage of the individual internal organizations or business divisions in your company, that is - chargeback. Thanks to the new feature, you can define groups and assign namespaces to these groups. Then, you can retrieve the license usage reports for the individual groups. As a result, you can analyze and understand the contribution of each group to the overall license usage, especially during the peak, or for any time range. You can also see which group has the highest and lowest usage, and what software is mostly used throughout your organization.

To filter the license usage by groups, divide your namespaces into groups by labeling them. You can assign multiple namespaces to one group. If you do not assign a namespace to any group, you do not track its contribution to the overall usage. Next, enable filtering feature in the `IBMLicensing` custom resource, and retrieve the license usage with the `/products` API with `groupName` parameter.

Filtering by groups

Note: The following procedure is dedicated to tracking the license usage of products that are licensed with Virtual Processor Core (VPC) and Processor Value Unit (PVU). For the products that are licensed with other license metrics, check the product documentation to check if filtering by groups is supported and on what conditions.

Complete the following procedures to enable filtering by group.

1. Define groups

To define groups, add the group labels to your namespaces by completing the following steps:

1. Log in to your OpenShift cluster console.
2. Go to **Administration** > **Namespaces**.

3. Find the namespace that you want to label.
4. Click the overflow action menu next to the namespace name, and select **Edit labels**.
5. Add the following label to the list:

   `ibm-chargeback-group=<groupName>`

   Where **groupName** is the name of the group that you will use to filter the data.

   Important: The name of the group must be other than **default**. This name is reserved for the development use only.
6. Repeat these steps for the remaining namespaces that you want to divide into groups. You can assign more than one namespace to one group.

2. Enable filtering by group

To enable filtering by group, complete the following steps.

1. Edit the **IBMLicensing** custom resource.
2. Add the following parameters to the **IBMLicensing** section, under **spec**:
   - To enable filtering by group, add the following line:

     `chargebackEnabled: true`
   - Optional: To set the retention period for which the filtered data is stored, add the following line and specify the number of days for storing the filtered data:

     `chargebackRetentionPeriod: <number of days>`

     Note: Specify the **chargebackRetentionPeriod** to define the number of days after which the data is deleted. If you do not add **chargebackRetentionPeriod** to the custom resource, the retention period is set to 62 days.

Retention period

Storing information about license usage of the products that are installed on the namespaces that belong to the groups requires extra memory. Because of that, the filtered data can only be stored for a limited time. By default, the **chargebackRetentionPeriod** is set to 62 days. In case of issues with retrieving the filtered data, decrease the retention period to avoid overflow.

Note: Remember to regularly generate audit snapshot to preserve all the data required for compliance purposes. Generate audit snapshot at least once during the span of retention period.

To estimate the retention period that best suits your environment consider the number of products that are deployed in the namespace, and the number of groups that you define.

Given that the maximal accepted size of the storage is 1 MiB, which equals 1048576 bytes, you can use the following formula to estimate the maximal retention period for your environment:

```
1048576 / (number of products * number of groups * 11.85)
```

The outcome is the number of days to which you should set the retention period for the specified number of products and groups.

The following table shows the estimated retention period for different number of products and groups expressed in the number of days:

Table. Estimated retention period

Environment	Small	Medium	Large
Products	up to 50	50-75	more than 75
Suggested retention period	1 quarter (90 days)	2 months (60 days)	1 month (30 days)

3. Retrieve license usage data by group

To retrieve the license usage data for the defined group, use the **/products** API and use the **groupName** parameter for filtering.

You can retrieve license usage for more than one group by adding more **groupName=<groupName>** parameters to get the total contribution of these groups.

API URL: **<License Service URL>/products?token=<token>&groupName=<groupName>**

For more information, see Retrieving license usage of products.

Understanding the reported values

The API retrieves the license usage of products that are deployed in the namespaces that are assigned to the defined group or groups. The license usage is expressed by the **metricQuantity** value which is the number of metric units that the group contributes to the overall usage of the product on the cluster. As a result the **metricQuantity** be expressed by a fraction.

Example scenario

You can use chargeback to track license usage for your products in different divisions in your organization. First, think about how many groups you can distinguish, and gather information about which namespaces these groups use.

In this example, the organization needs to define two groups **HR** and **Procurement**. The organization follows the procedure and defines the groups by adding the labels to appropriate namespaces, and then enables filtering by group. As the organization defines only 2 groups that use only one product, the chargeback retention period is set to 90 days as recommended for small environments.

- The following API retrieves license usage for the **HR** group:

  LS-host/products?token=token&groupName=HR

  Response:

  ```
  "name": "IBM Cloud Pak for Integration",
  "id": "c8b82d189e7545f0892db9ef2731b90d",
  "metricPeakDate": "2021-07-08",
  "metricName": "VIRTUAL_PROCESSOR_CORE",
  "metricQuantity": 1.5
  ```

- The following API retrieves license usage for the **Procurement** group:

  LS-host/products?token=token&groupName=Procurement

  Response:

  ```
  "name": "IBM Cloud Pak for Integration",
  "id": "c8b82d189e7545f0892db9ef2731b90b",
  "metricPeakDate": "2021-07-08",
  "metricName": "VIRTUAL_PROCESSOR_CORE",
  "metricQuantity": 0.5
  ```

Based on the example, the **HR** group contributes 1.5 VPCs of the overall usage of IBM Cloud Pak for Integration on the peak date, while **Procurement** group contributes 0.5 VPC.

- The following API retrieves the overall license usage that is registered on a cluster:

  LS-host/products?token=token

  Response:

  ```
  "name": "IBM Cloud Pak for Integration",
  "id": "c8b82d189e7545f0892db9ef2731b90d",
  "metricPeakDate": "2021-07-08",
  "metricName": "VIRTUAL_PROCESSOR_CORE",
  "metricQuantity": 4
  ```

As the overall license usage of IBM Cloud Pak for Integration on the peak date equals 4, and the **HR** group contributes 1.5 VPCs and the **Procurement** group 0.5 VPC, you can calculate that the remaining usage, 2 VPCs, is used outside of the defined groups.

The following image represents the license usage on the cluster.

License usage of Cloud Pak for Integration

Metric quantity = **4 VPCs**

Group 1 **HR**	Group 2 **Procurement**	Outside of defined groups
Reported bundled products: IBM MQ Advanced vCPU= 1 Metric conversion: 2:1 Converted metric contribution = 0.5 VPC IBM MQ vCPU = 1 Metric conversion: 4:1 Converted metric contribution = 1 VPC	Reported bundled products: IBM MQ Advanced vCPU = 1 Metric conversion: 2:1 Converted metric contribution = 0.5 VPC	Reported bundled products: IBM MQ Advanced vCPUs = 2 Metric conversion: 2:1 Converted metric contribution = 1 VPCs IBM API Connect vCPU= 1 Metric conversion = 1:1 Converted metric contribution = 1 VPC
Group contribiution: **1.5 VPCs**	Group contribution: **0.5 VPCs**	Contribution: **2 VPCs**

The following statements summarize the license usage calculations for the cluster:

- The group contribution, that is expressed in the number of metrics, is a sum of converted metric contributions of all bundled products that are reported for this group.
- The converted metric contribution of a bundled product in the group is proportional to the overall use of the product on the cluster.
- To calculate the contribution of the `HR` group, the converted metric contribution needs to be estimated for IBM MQ Advanced and IBM MQ.
- The following example shows how to calculate the converted metric contribution for IBM MQ Advanced that is deployed within the `HR` group.

Table. Example metric contribution for IBM MQ Advanced

Information	Value	Comments
Overall usage of IBM MQ Advanced on the cluster	1 vCPU + 1 vCPU + 2 vCPUs = 4 vCPUs	To calculate the overall usage of the bundled product, add the vCPUs that are used by all the instances of the bundled product in the cluster. vCPU is calculated according to the IBM Container Licensing rules. For more information, see IBM Container Licenses on Passport Advantage.
Metric conversion for IBM MQ Advanced	2:1	Metric conversion is defined in the licensing document of the product
Metric converted quantity for IBM MQ Advanced on the cluster	4 vCPUs * 2:1 = 2 VPCs	To calculate the metric converted quantity, multiply the overall usage of the bundled product on the cluster by the metric conversion ratio. As a result, the value represents the number of VPCs that the bundled product contributes on the cluster
Converted metric contribution of IBM MQ Advanced in the `HR` group	1 vCPU /4 vCPUs * 2 VPCs = 0.5 VPC	To calculate the contribution of the bundled product within the group divide the number of the vCPUs used within the group by the overall number of vCPUs used on the cluster and multiply it by the metric converted quantity for the cluster.

You can base on the example to similarly calculate the converted metric contribution of any bundled product within the group.

Enabling hyperthreading

This feature is available from IBM Cloud Pak® foundational services version 3.11.x.

License Service supports hyperthreading, also referred to as Simultaneous multithreading (SMT) or Hyper-Threading (HT). To take advantage of the benefits of using this technology, enable hyperthreading in License Service. Hyperthreading might have great impact on your licensing costs. For more information, see Hyperthreading.

You can enable hyperthreading for the clusters where all working nodes have SMT or HT enabled. To enable hyperthreading, complete the following steps.

1. Edit the `IBMLicensing` custom resource.
2. Add the following lines to the `IBMLicensing` section, under `spec`:

```
features:
  hyperThreading:
    threadsPerCore: <number of threads>
```

Note: Set the value of the `threadsPerCore` parameter to the lowest number of threads per core in your infrastructure. The following table lists the acceptable values.

Table. Acceptable values for threads per core

Value	Description
`1`	Hyperthreading is not enabled.
`2`	HT and SMT2 is enabled with the declared number of 2 threads per core.
`4`	SMTP4 is enabled with the declared number of 4 threads per core.
`8`	SMTP8 is enabled with the declared number of 8 threads per core.

Viewing licensing information on Administration panel

You can view the license usage information for your cluster on the Administration panel of your IBM Cloud Pak console.

This feature is available from IBM Cloud Pak® foundational services version 3.5.x.

Licensing on Administration panel

The IBM Cloud Pak® foundational services console includes an Administration panel that provides Cluster Administrators an at a glance overview of a cluster.

One of the summary cards on Administration panel is **Licensing**.

You can monitor your licensing statistics on the **Licensing** summary card within the *System overview* section of the Administration panel dashboard when you log in as a Cluster Administrator or Cloud Pak Administrator . After you log in, access the main navigation menu and click **Home** to access this dashboard.

This summary card displays the following information:

Table. Information description

Information	Description
IBM pods missing license reporting	The number of pods that have product annotations, however, because the product annotations are incomplete or improper, License Service cannot measure their license usage. Click **View details** to see the names of the pods. The pods with invalid annotations are listed in the audit snapshot. For audit purposes, all pods should be reported or accounted for. The pod name might identify the associated software. Analyze the pods, and decide whether you need to fix the annotations. If the software requires License Service reporting, contact the support of the product. If the software does not require License Service reporting, for example, it is free of charge, you can decide not to fix the annotations.
Total products reported	The number of products that License Service detects on the cluster and calculates license usage for. Click **View details** to see the names of the products and their license usage. The license usage that is listed is the most current license usage collected from the cluster.

If you deploy License Service Reporter, the summary card additionally shows a **View license usage** link. The link leads to the License Service Reporter user interface that presents the highest license usage of your products within the reporting period for a multicluster environment.

Retrieving license usage data

You can retrieve license usage data per cluster from License Service and for multiple clusters from License Service Reporter through dedicated API calls.

- Per cluster from License Service
- For multiple clusters from License Service Reporter

Verifying completeness of license usage data and troubleshooting

Learn how to check whether License Service is properly deployed to your cluster and whether it collects the complete license usage data from your cluster. View the steps for troubleshooting most common problems.

- Validating License Service deployment
- Enabling additional information in logs for troubleshooting purposes
- Checking completeness of license usage data
- Interpreting the warnings
- Viewing information about the pods from which license usage cannot be collected

Common problems

- License Service instances are crashing with `CrashLoopBackOff` status
- Duplicated products when retrieving license usage with `/products` API
- License Service API is unavailable with 503 Service Unavailable error
- Multiple IBM Licensing instances

Validating License Service deployment

License Service is automatically deployed with IBM Cloud Pak for AIOps. You can check whether License Service was successfully deployed on the cluster by using any available tools. For example, you can log in to the cluster and run the following command:

```
kubectl get pods --all-namespaces | grep ibm-licensing | grep -v operator
```

The following response is a confirmation of successful deployment:

```
 1/1     Running
```

Enabling additional information in logs for troubleshooting purposes

By default, the License Service instance pod logs contain only the basic information about the service. You can enable additional information in logs for troubleshooting purposes by updating the `OperandConfig`. Add the `logLevel` parameter to the `ibm-licensing operator` section, under `spec`.

Note: You can access the `OperandConfig` by using the Red Hat® OpenShift® Container Platform console or by using the command-line interface (CLI).

1. Access the `OperandConfig`.

    - In Red Hat® OpenShift® Container Platform console, go to **Installed Operators** > **Operand Deployment Lifecycle Manager** > **OperandConfig** tab. Edit the `common-service OperandConfig` instance that the IBM Cloud Pak

foundational services operator already created for you. Click the overflow menu icon, and then click **Edit OperandConfig**.

- Use the following command:

  ```
  oc edit OperandConfig common-service -n ibm-common-services
  ```

2. Add the following parameter to the `ibm-licensing operator` section, under `spec`.

   ```
   - name: ibm-licensing-operator
     spec:
       IBMLicensing:
         logLevel: <option>
   ```

   The available `logLevel` options:

   - `DEBUG` - This option enables all debug information in logs.
   - `VERBOSE` - This option extends the logs with information about license calculation and API calls.

Checking completeness of license usage data

License Service runs health checks every day. The service scans the cluster, checks the validity of your pods and the underlying metadata and whether the data is properly imported and processed. If any information about IBM Cloud Pak for AIOps, or its bundled product that is crucial for license measurements is missing, a warning appears in the logs.

Use a logging service, for example ELK stack or other third-party solution to monitor the logs. If you use ELK stack, you can filter the dashboard by the following tags to see the License Service warnings: `MissingPodAnnotation` or `DataImportProblem`.

Note: It is a good practice to check the logs regularly, and whenever you notice that a product is not being reported.

Interpreting the warnings

License Service generates two types of warnings.

Missing pod annotations

The warnings from the service inform you when a pod is not properly annotated. Improper annotations result in the license usage of a product not being measured. The annotations include the product metadata, such as the product ID and metric, and are implemented by the IBM team that develops the product to enable license usage measurements.

Example - Missing pod annotations warning

Each warning contains a timestamp, information about a problematic pod, and a list of missing annotations. The following example shows a warning in the logs:

```
2020-Feb-11 12:38:59.597 [scheduling-1] WARN  [MissingPodAnnotation]
Pod kube-system/iam-onboarding-mktmv has invalid annotation:
productChargedContainers,productMetric
```

Resolving the warnings

To resolve this warning, identify the software which is deployed on the pod that is listed in the warning. Contact IBM Support for the identified software and provide them with the text of the warning. The product team needs to improve or add annotations to the problematic product so that it can be reporter by License Service. License Service support cannot resolve this issue.

Problems with importing or processing data

The warnings from the service inform you when the data might not be imported or processed by License Service. More information is listed in the logs and depends on the nature of the problem.

Example - Problems with importing or processing data

```
2020-03-04 10:52:55.889 [scheduling-1] ERROR [DataImportProblem] Error while importing and processing data
```

Viewing information about the pods from which license usage cannot be collected

You can check the list of pods from which License Service might not collect license usage in the **unrecognized_apps.csv** file that is retrieved with an audit snapshot. For more information, see Retrieving an audit snapshot.

License Service instances are crashing with `CrashLoopBackOff` status

If your License Service pods are crashing and you see multiple instances of License Service with the **CrashLoopBackOff** status in your OpenShift console, you might have License Service deployed to more than one namespace. As a result, two License Service operators are running in two namespaces and the service crashes. The **ibm-licensing-operator** should only be deployed in the **ibm-common-services** namespace, however, if you deployed License Service more than once, the older version of License Service might be deployed to **kube-system** namespace.

Complete the following steps to fix the problem:

1. To check whether the **ibm-licensing-operator** is deployed to **kube-system** namespace, run the following command:

   - **Linux®:** `kubectl get pod -n kube-system | grep ibm-licensing-operator`
   - **Windows:** `kubectl get pod -n kube-system | findstr ibm-licensing-operator`
2. If the response contains information about the running pod, uninstall License Service from **kube-system** namespace. To uninstall License Service in OpenShift console, complete the following steps:

   a. Log in to the OpenShift console.

   b. Go to **Operators > Installed Operators**.

   c. Switch to the **kube-system** project.

   d. Find the **IBM Licensing Operator**, click the overflow menu icon and select **Uninstall Operator**.

Duplicated products when retrieving license usage with `/products` API

If different bundled products report license usage under the same product but with a different version, you might get duplicated products when you run the **/products** API.

This is a known issue that is resolved with the 3.7.2 release of IBM Cloud Pak foundational services. Upgrade to the latest version of IBM Cloud Pak foundational services to fix this issue.

Note: By default the **/products** query returns the data for the last 30 days. Because of that, after you upgrade to the newest version of IBM Cloud Pak foundational services you might still get duplicated products for the days prior to the upgrade. You can mitigate this issue by filtering the data and setting the **startDate** and **endDate** parameters to view the usage for the specific date.

License Service API is unavailable with 503 Service Unavailable error

You might get 503 Service Unavailable error when you make a License Service API call when you use the custom ingress certificate. The custom ingress certificate is not acceptable for License Service. To fix this issue, complete the following actions:

1. Generate the correct certificate for Fully Qualified Domain Name (FQDN) of License Service. To learn how to check the License Service URL, see Obtaining the License Service URL.

2. Configure your custom certificate.

Multiple IBM Licensing instances

Licensing Service as a singelton doesn't support running multiple instances in the same environment simultaneously. In case if multiple IBM Licensing instances are present, then only the first-formed object (judged by creation timestamp) will be used to create a Licensing Service instance.

When performing the upgrade on the environment, if there are multiply instances, they will be preserved along with active Licensing Service Services, but Custom Resource will be annotated **`INACTIVE`**. It is recommended to remove CR to avoid incoherent behaviour of the Licensing operator.

To display currently running instances, run the following command:

```
oc get IBMLicensing
```

The sample output:

```
NAME         POD PHASE
instance2    Running
instance3
instance4
```

Only the **`instance2`** in **`Running`** Phase which means **`instance2`** is the only that is **`ACTIVE`**.

To resolve this issue, delete the **`INACTIVE`** instances by using following command:

```
oc delete IBMLicensing <instance name>
```

Where, **instance name** is the name of **`INACTIVE`** CR. In case of this example, the **instance name** would be **`instance3`** and **`instance4`**.

Using custom certificates

Learn how to use a custom certificate for License Service API over https.

To configure a custom certificate for License Service communication, complete the following steps:

1. Change the certificate name to **`tls.crt`**.
2. Change the name of the key to **`tls.key`**.
3. Run the following command to change the directory to where the certificate and the key are stored:

   ```
   cd <certificate_directory>
   ```

4. Create a secret by using the following command:

   ```
   licensingNamespace=$(oc get pods --all-namespaces | grep "ibm-licensing-service-" | awk
   {'print $1'})
   kubectl create secret tls ibm-licensing-certs --key tls.key --cert tls.crt -n
   ${licensingNamespace}
   ```

5. Edit the IBMLicensing custom resource to include the certificate. Add the following parameters to the **`IBMLicensing`** section, under **`spec`**:

   - To enable the **`https`** connection, add the following line:

     ```
     httpsEnable: true
     ```

   - To apply the custom certificate that you created in step 4 as **`ibm-licensing-certs`**, add the following line:

     ```
     httpsCertsSource: custom
     ```

     For example:

     ```
     apiVersion: operator.ibm.com/v1alpha1
     kind: IBMLicensing
     ```

```
metadata:
   name: instance
spec:
   httpsEnable: true
   httpsCertsSource: custom
```

When you change or replace a Custom Certificate to get the new Certificate, applied to the License Service, make sure to delete the **ibm-licensing-operator-xxxx** pod. This operation will force restart of License Service with updated certificate.

Tracking license usage in a multicluster environment

For better understanding of license usage in your environment, you can track the license usage of your products across multiple clusters. You can manually prepare a consolidated report, or deploy and configure License Service Reporter.

Audit readiness in a multicluster environment

License Service is an automatic solution that collects and measures the license usage of your software that is needed for audit. No manual post-processing is required.

In case of a multicluster environment, you only need to generate audit snapshots for each cluster for auditors to verify the overall license compliance. You do not need a consolidated snapshot for all your clusters.

Consolidating license usage data for tracking purposes

If you want to have better visibility of the license usage across multiple clusters, you can consolidate data on a single report. You can use one of the following tools for that purpose:

- Deploy and configure License Service Reporter, and monitor the usage on a dedicated dashboard. For more information, see Tracking license usage in multicluster environment with License Service Reporter.
- For smaller environments, or if you do not have resources for deploying and configuring License Service Reporter, you can manually prepare a consolidated report. For more information, see Manually tracking license usage in multicluster environment.

Determining the highest license usage for multiple clusters

The highest license usage in a multicluster environment is the highest daily sum of used metric units by a product on all clusters within the reporting period.

For IBM Cloud Paks®, the highest license usage is counted for the entire IBM Cloud Pak, not the individual bundled products.

The highest license usage across multiple clusters is automatically calculated by License Service Reporter and based on data delivered by your clusters, and displayed on a dashboard. You can also calculate it manually.

The logic behind determining the highest license usage for multiple clusters

First, the daily peaks from multiple clusters are collected.

Example: The following charts show the license usage of Product A that is licensed with Virtual Processor Core (VPC) metric and runs on Cluster 1 and Cluster 2.

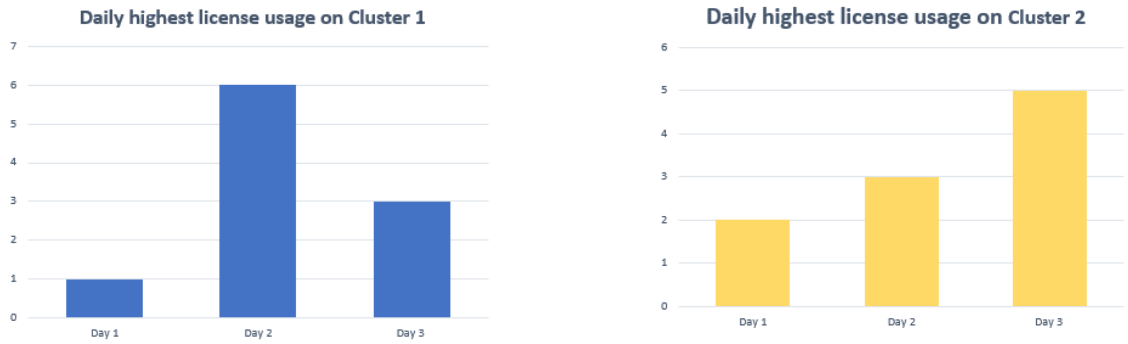


Figure. FIPs overview

Next, the daily peaks from clusters are added. The highest sum is the highest license usage in the environment in the reporting period.

Example: The following chart shows the sums of daily peaks for Product A that was registered for the period of 3 days on Cluster 1 and Cluster 2.

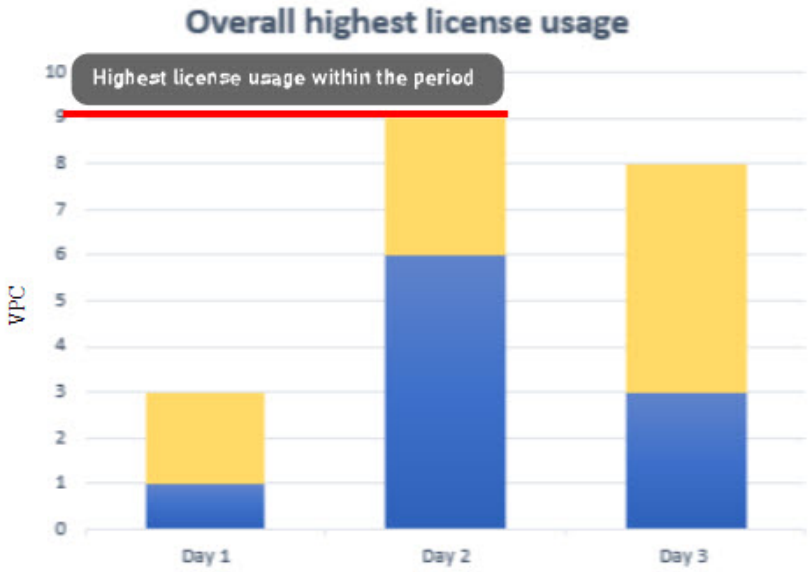


Figure. FIPs overview

Tracking license usage in multicluster environment with License Service Reporter

License Service Reporter is an extension of License Service that aggregates the license usage data from multiple clusters. The collective highest license usage is presented on the user interface.

Retrieving license usage in the multicluster environment with License Service Reporter

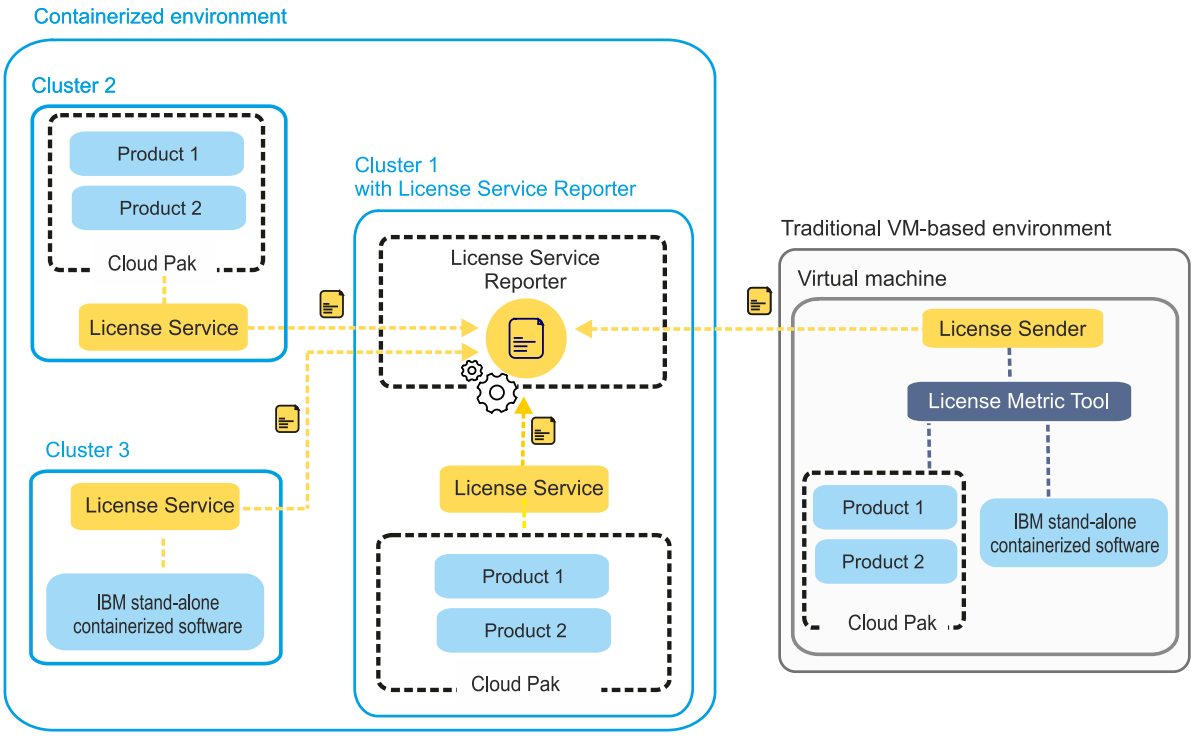


Figure. Multicluster environment data flow

License Service Reporter

License Service Reporter is an aggregator that adds and analyzes the license usage data from multiple clusters, and retrieves the highest license usage of IBM Cloud Paks and IBM® stand-alone containerized software. Deploy one central License Service Reporter on a selected cluster. License Service Reporter uses extra resources for aggregation.

Currently, License Service Reporter must be deployed in the same namespace as License Service. License Service Reporter is not supported in multitenancy scenario, where License Service is deployed in dedicated control plane.

Supported platforms

License Service Reporter is only supported on Linux® x86_64 on Red Hat® OpenShift® Container Platform.

Data sources

- First data sent to the License Service Reporter covers 30 day period back form the day you configure License Service instance as a data source.
- To deliver data to License Service Reporter from multiple clusters, configure the License Service instance for each cluster. Enable License Service to connect and deliver the license usage data from the cluster to the central License Service Reporter.
- Some IBM Cloud Paks products, and IBM stand-alone containerized software can be deployed on clusters and in the traditional VM-based environment. To correctly aggregate license usage for these products, License Service Reporter needs to get data from clusters and virtual machines. The license usage data from virtual machines is reported by IBM License Metric Tool. Use License Sender, an integration tool, to deliver license usage data from License Metric Tool to License Service Reporter. For more information, see Delivering license usage of IBM Cloud Pak solutions installed in non-containerized environment from License Metric Tool to License Service Reporter.

Network flow

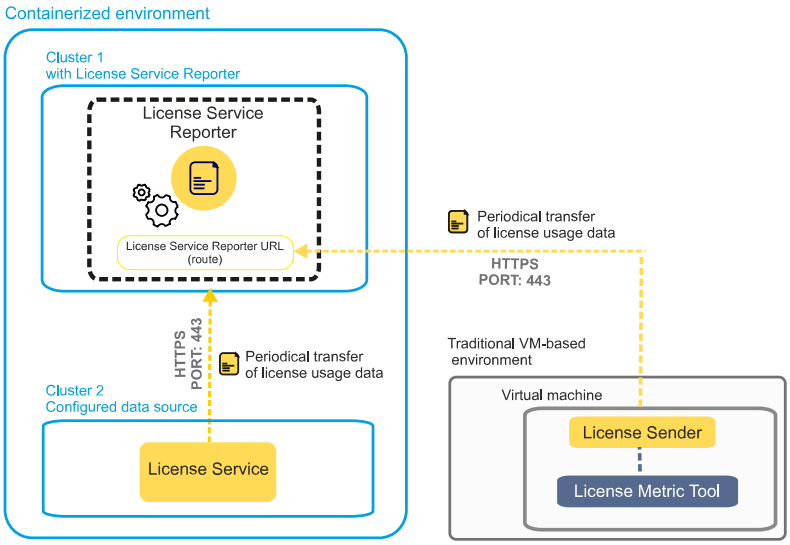


Figure. Network flow

When you configure the data source to report to License Service Reporter, the following actions are performed:

- In containerized environments, License Service instance on a cluster activates the sender module. The module periodically connects to the License Service Reporter URL and pushes the license usage data.
- In traditional VM-based environment, the License Sender component periodically connects to the License Service Reporter URL and pushes the license usage data.

The sender connects via HTTPS and so the 443 port should be open.

The License Service Reporter URL is defined in the **ibm-license-service-reporter** route in OpenShift console of the cluster where you deployed License Service Reporter. This URL is a hostname of License Service Reporter that should be used by the network administrators for the firewall setup. For more information, see Obtaining License Service Reporter URL.

Getting started with License Service Reporter

Complete the following actions to track license usage in multicluster environment with License Service Reporter:

- Deploying License Service Reporter
- Configuring the data sources
- Validating License Service Reporter deployment
- Viewing license usage on the Licensing dashboard
- Generating audit snapshot in License Service Reporter

Deploying License Service Reporter

Deploy License Reporter on a cluster and use it as an aggregator to view the collective license usage of your products that are deployed on multiple clusters. You can deploy License Reported by using the OpenShift console or the CLI.

Supported platforms

License Service Reporter is only supported on Linux® x86_64 on Red Hat® OpenShift® Container Platform.

Hardware requirements

- **CPU limits**: 900m
- **Memory limits**: 896Mi

Prerequisites

Deploy License Service Reporter on the OpenShift-managed cluster where the IBM Cloud Paks® are deployed.

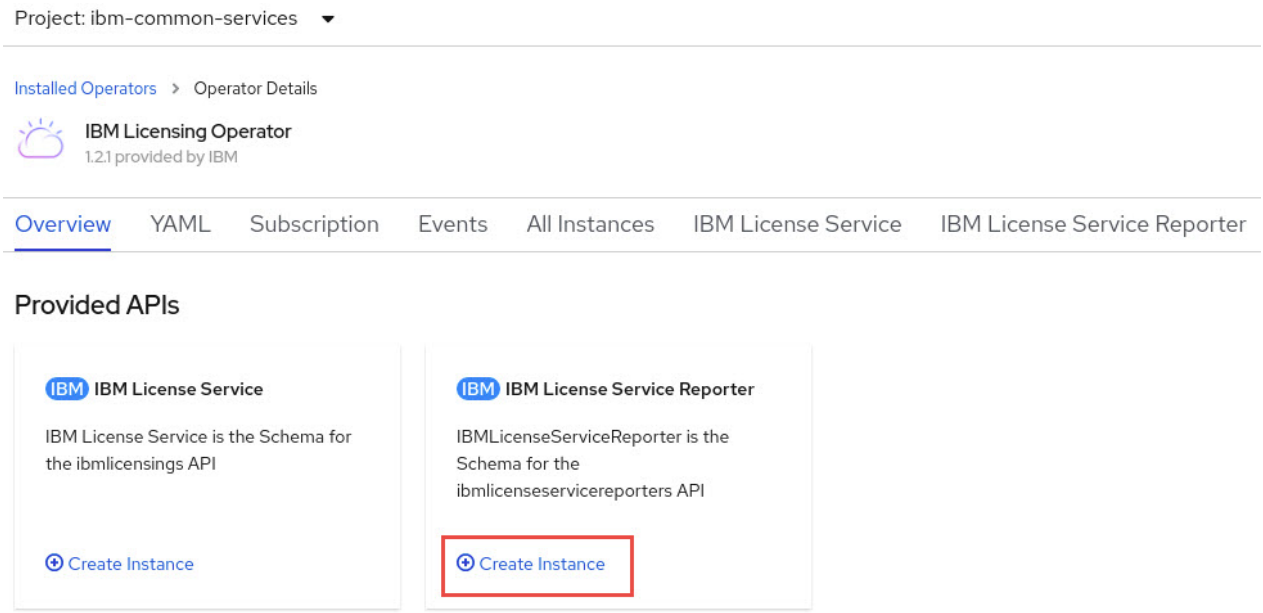
Deployment

Use one of the following procedures to deploy License Service Reporter on a cluster.

- Deploying License Service Reporter by using the OpenShift console
- Deploying License Service Reporter by using the CLI

Deploying License Service Reporter by using the OpenShift console

1. Log in to your OpenShift cluster console.
2. Go to **Operators** > **Installed Operators**.
3. Set the project to **All Projects**.
4. Find and select the **IBM Licensing Operator**.
5. On the **IBM License Service Reporter** tile, click **Create instance**.



6. Review the contents of the editor. Make sure that the definition contains the following information. If not, paste the following content into the editor, and click **Create**.

```
apiVersion: operator.ibm.com/v1alpha1
kind: IBMLicenseServiceReporter
metadata:
   labels:
     app.kubernetes.io/instance: ibm-licensing-operator
     app.kubernetes.io/managed-by: ibm-licensing-operator
     app.kubernetes.io/name: ibm-licensing
   name: instance
   namespace: ibm-common-services
```

Note: License Service Reporter uses the default storage class. To use a different storage class, add and define the 'storageClass' parameter in the `spec` section. For example:

```
spec:
   storageClass: storage-class-internal
```

7. Go back to **Operators** > **Installed Operators**.
8. Find and select **Operand Deployment Lifecycle Manager**.
9. Go to the **OperandConfig** tab, and click the **common-service** instance.
10. Find the **ibm-licensing-operator** and add **IBMLicenseServiceReporter: {}** to its **spec** section. For example:

```
- name: ibm-licensing-operator
    spec:
      IBMLicenseServiceReporter: {} # <- this needs to be added
      IBMLicensing:
        datasource: datacollector
        resources:
          limits:
            cpu: 300m
            memory: 850Mi
          requests:
            cpu: 200m
            memory: 430Mi
      operandBindInfo: {}
```

11. To apply the changes, click **Save**.

Deploying License Service Reporter by using the CLI

1. Create the `license-reporter.yaml` file with the following content:

```
apiVersion: operator.ibm.com/v1alpha1
kind: IBMLicenseServiceReporter
metadata:
   labels:
     app.kubernetes.io/instance: ibm-licensing-operator
     app.kubernetes.io/managed-by: ibm-licensing-operator
     app.kubernetes.io/name: ibm-licensing
   name: instance
   namespace: ibm-common-services
```

   Note: License Service Reporter uses the default storage class. To use a different storage class, add and define the 'storageClass' parameter in the `spec` section. For example:

```
spec:
   storageClass: storage-class-internal
```

2. Run the following command:

```
oc apply -f license-reporter.yaml
```

3. Edit the **OperandConfig** custom resource. To include License Service Reporter, run the following command:

```
oc edit OperandConfig common-service -n ibm-common-services
```

4. Search for **ibm-licensing-operator** with the following command and click **Enter**:

```
?ibm-licensing-operator
```

5. Type **i** and edit the **spec** section of the **ibm-licensing-operator** by adding **IBMLicenseServiceReporter: {}**. For example:

```
- name: ibm-licensing-operator
  spec:
    IBMLicenseServiceReporter: {}
    IBMLicensing:
```

6. Click **Esc** and type **:x** to save your changes.
7. Optional: If you have issues with applying the changes, save, change and update the yaml.
    1. Save the .yaml into a file by running the following command.

```
oc get OperandConfig common-service -n ibm-common-services -o yaml > opconfig.yaml
```

2. Add **IBMLicenseServiceReporter: {}** under the **ibm-licensing-operator spec** section.
3. Apply the yaml with the following command:

```
oc apply -f opconfig.yaml
```

Configuring data sources

After deploying License Service Reporter, configure the License Service instance to enable data feeds from your clusters to License Service Reporter.

Complete the following actions to configure the following data sources to deliver licensing data to License Service Reporter.

- Configuring the License Service instance for the cluster where License Service Reporter is deployed
- Configuring the License Service instance for the remaining OpenShift clusters
- Configuring the License Service instance for non-OpenShift clusters
- Delivering license usage of software installed in non-containerized environment from License Metric Tool to License Service Reporter
- Uploading audit snapshots from offline environments into License Service Reporter
- Verifying the License Service connection to the License Service Reporter

Configuring the License Service instance for the cluster where License Service Reporter is deployed

For the cluster where you deployed License Service Reporter, the License Service instance is automatically configured. Thanks to that, you do not need to perform any configuration and the license usage information are delivered to License Service Reporter.

Optionally, you can configure the License Service instance if you want to set the custom cluster name and ID. By default, the cluster name is set to the name of the first master node, and an ID is set to the **`kube-system`** namespace UID.

Important: Cluster ID and cluster name are included in the audit snapshot and must be clear for an auditor to identify the cluster.

Restriction: Cluster ID and cluster name cannot be changed after you configure the instance.

To set the custom cluster name and cluster ID, complete the following steps.

1. Log in to your OpenShift cluster console.
2. Go to **Operators** > **Installed Operators**.
3. Find and select the **IBM Licensing Operator**.
4. Go to the **IBM License Service** tab.
5. Edit the instance by expanding the menu at the end of the line and selecting **Edit IBMLicensing**.
6. In the YAML, add the following parameters and their values in the **`spec`** section under **`sender`** and click **Save**.

   ```
   clusterID: <cluster_id>
   clusterName: <your_cluster_name>
   ```

   Where

   - **`<cluster_id>`** is a custom value expressed by a unique alphanumeric string that defines the cluster.
   - **`<cluster_name>`** is a custom, unique name that identifies the cluster on the License Service Reporter user interface.

Configuring the License Service instance for the remaining OpenShift clusters

Configure the License Service instance to enable data feeds from for all remaining clusters to License Service Reporter for multi-cluster aggregations. Configure all clusters with software that should be included in the licensing calculations.

1. Copy the secret with a License Service Reporter token from the cluster where License Service Reporter is deployed.
    1. Log in to the OpenShift console of the cluster where you deployed License Service Reporter.
    2. Go to **Workloads** > **Secrets** and select the project where License Service Reporter is installed.
    3. Find and select **ibm-licensing-reporter-token**.
    4. Go to the **YAML** tab and copy the entire YAML with the secret.
2. Create the secret with a token on the cluster from which you want to retrieve licensing data.
    1. Log in to the OpenShift cluster console.
    2. Go to **Workloads** > **Secrets**.
    3. Expand the **Create** menu and select **From YAML**.
    4. Paste the contents of the YAML that you copied in step 1d into the editor, delete `ownerReferences` section and click **Create**.
3. Modify the License Service instance to enable data feeds from the cluster to License Service Reporter.
    1. Log in to your OpenShift cluster console.
    2. Go to **Operators** > **Installed Operators**.
    3. Find and select the **IBM Licensing Operator**.
    4. Go to the **IBM License Service** tab.
    5. Edit the instance by expanding the menu at the end of the line and selecting **Edit IBMLicensing**.
    6. Add the following content to the `spec` section in the YAML and click **Save**.

        ```
        sender:
          reporterSecretToken: ibm-licensing-reporter-token
          reporterURL: <reporter_url>
          clusterID: <cluster_id>
          clusterName: <your_cluster_name>
        ```

        Where

        - `<reporter_url>` is a URL of the License Service Reporter that you deployed. To get the URL, complete the following steps:
            - Log in to the OpenShift console of the cluster where you deployed License Service Reporter.
            - Go to **Networking** > **Routes** and select the **ibm-common-services** project.
            - Find **ibm-license-service-reporter** and copy the link that is provided under Location.

Note: From IBM Cloud Pak foundational services version 3.6.x, the `clusterID` and `clusterName` parameters are optional. If you do not need to customize the ID and name of the cluster, remove the parameter from the `sender` section. Cluster ID and cluster name are included in the audit snapshot and must be clear for an auditor to identify the cluster.

Restriction: Cluster ID and cluster name cannot be changed after you configure the instance.

- `<cluster_id>` - By default, the cluster ID is set to the `kube-system` namespace UID. To set a custom ID, include this parameter in the YAML and provide a custom value expressed by a unique alphanumeric string that defines the cluster.
- `<cluster_name>` - By default, the cluster name is set to the name of the first master node. To set a custom cluster name, include the parameter in the YAML and provide a custom, unique name that identifies the cluster on the License

Service Reporter user interface.

Configuring the License Service instance for non-OpenShift clusters

You can configure License Service instance for non-OpenShift Kubernetes clusters to deliver data to License Reporter under following circumstances:

- In addition to IBM Cloud Paks, you have IBM stand-alone containerized software that is deployed on non-OpenShift Kubernetes clusters.
- Your IBM stand-alone containerized software is enabled for reporting and has integrated License Service.
- You want to track license usage of your IBM Cloud Paks and IBM stand-alone containerized software on one report.

To configure the License Service instance for non-OpenShift Kubernetes clusters, complete the following steps.

1. Copy the secret with the License Service Reporter token from the cluster where License Service Reporter is deployed.
    1. Log in to the cluster where you deployed License Service Reporter and run the following command.

        ```
        kubectl get secret ibm-licensing-reporter-token -o yaml -n ibm-common-services > reporter-token.yml
        ```

    2. Open **reporter-token.yml** in the text editor and remove the **ownerReferences** section.
2. Create the secret with a token on the cluster from which you want to retrieve license usage data. Log in to the cluster and run the following command to create a secret.

    ```
    kubectl apply -f reporter-token.yml -n ibm-common-services
    ```

3. Modify the License Service instance to enable data feeds from the cluster to License Service Reporter.
    1. Run the following command.

        ```
        kubectl edit IBMLicensing -o yaml -n ibm-common-services
        ```

    2. Modify the **.yaml**, add the following content under 'spec' section and save the file.

        ```
        sender:
          reporterSecretToken: ibm-licensing-reporter-token
          reporterURL: <reporter_url>
          clusterID: <cluster_id>
          clusterName: <cluster_name>
        ```

        Where

        - **<reporter_url>** is a URL of the License Service Reporter that you deployed. To get the URL, complete the following steps:
            - Log in to the OpenShift console of the cluster where you deployed License Service Reporter.
            - Go to **Networking** > **Routes** and select the **ibm-common-services** project.
            - Find **ibm-license-service-reporter** and copy the link that is provided under Location.

        Note: You can also use the following command: **kubectl get routes -n ibm-common-services | grep ibm-license-service-reporter | awk '{print $2}'**

    Note: From IBM Cloud Pak foundational services version 3.6.x, the **clusterID** and **clusterName** parameters are optional. If you do not need to customize the ID and name of the cluster, remove the parameter from the **sender** section. Cluster ID and cluster name are included in the audit snapshot and must be clear for an auditor to identify the cluster.

    Restriction: Cluster ID and cluster name cannot be changed after you configure the instance.

    - **<cluster_id>** is an optional parameter. By default, the cluster ID is set to the **kube-system** namespace UID. To set a custom ID, include this parameter in the YAML and provide a custom value expressed by a unique alphanumeric string that defines the cluster.
    - **<cluster_name>** is an optional parameter. By default, the cluster name is set to the name of the first master node. To set a custom cluster name, include the parameter in the YAML and provide a custom, unique name that

identifies the cluster on the License Service Reporter user interface.

Delivering license usage of software installed in non-containerized environment from License Metric Tool to License Service Reporter

For more information, see Delivering license usage of IBM Cloud Pak solutions and stand-alone containerized software installed in non-containerized environment from License Metric Tool to License Service Reporter.

Uploading audit snapshots from offline environments into License Service Reporter

This feature is available from IBM Cloud Pak® foundational services version 3.21.x.

Prerequisites

- License Service Reporter in version 1.18 or higher included in foundational services 3.21 and newer
- Audit snapshot must be generated by using License Service version 1.18 or higher included in foundational services 3.21 and newer

Note: If upload fails with error 422, make sure that the audit snapshot is generated from the correct License Service version, and that you use License Service Reporter that supports the upload.

If you use License Service to track license usage in offline environments, you can upload audit snapshot that is generated for offline environments into License Service Reporter. Thanks to the upload, the additional measurements are included in the collective license usage statistics on Licensing dashboard and in License Service Reporter reports.

Procedure

Complete the following actions to upload audit snapshots from offline environments into License Service Reporter.

1. Generate an audit snapshot from the offline environment. For more information and instructions, see Retrieving an audit snapshot.
2. Upload the generated audit snapshot **.zip** package into License Service Reporter with the following **CURL** query:

   ```
   curl -k -X PUT -F 'file=@<audit_snapshot_file_name.zip>' "<License Service Reporter URL>/snapshot?token=<token>"
   ```

   Where:

   <audit_snapshot_file_name.zip> is the name of your audit snapshot package

   <token> is your authentication token for connecting to License Service Reporter. For more information, see Obtaining and updating an API token.

   <License Service Reporter URL> is the URL of your License Service Reporter instance. For more information, see Obtaining License Service Reporter URL.

   For more information about this API, see License Service Reporter Swagger API schema.

   Note: The audit snapshot can cover data from any period, however the package cannot exceed 1 MB. If your package contains large amounts of data, upload takes long and you are notified that the connection being prematurely closed, change your load balancer settings.

   During the upload, License Service Reporter verifies the audit snapshot checksums and the signature to ensure data authenticity.

Verification

If upload completes successfully, you get the **HTTP** 200 response with the following message: **Audit snapshot file upload finished successfully**.

Changing load balancer settings

If your package contains large amounts of data, you might need to increase load balancer timeout to receive correct response from the server. Otherwise, upload times out with an error message that the client prematurely closed the connection while reading the response header. The timeout occurs when the process does not finish within the timeout period that is currently set for load balancer.

Note: If load balancer closes connection with License Service Reporter due to exceeded timeout, it does not always mean that the upload failed as the process continues anyway. Check logs to verify whether the upload was successful or not.

For example, if you have the HAProxy load balancer, complete the following steps to change the client and server timeout value:

1. On the load balancer node, edit the `/etc/haproxy/haproxy.cfg` file. In the defaults section, change the values for timeout client and timeout server from 1m to 5m.
2. Run the following command to restart HAProxy: `systemctl restart haproxy`.

Verifying the License Service connection to the License Service Reporter

Note: Data sent by License Service (License Service Reporter data source) will be visible in License Service Reporter after around an hour of configuring the data sources.

After you deploy the License Service instance successfully, you can verify whether the License Service has connected to License Service Reporter and sent data. After around 10 minutes of deploying License Services instance, check the License Service logs and look for the following similar entries:

```
2022-09-14 12:17:46.840 [scheduling-1] INFO  Starting sender task
2022-09-14 12:17:46.852 [scheduling-1] INFO  License Service Reporter URL = https://console-
openshift-console.apps.ls-js.cp.myibm.com, cluster name = my_cluster, cluster ID =
my_cluster_id
2022-09-14 12:17:46.934 [scheduling-1] INFO  Connected to License Service Reporter 1.18.0
2022-09-14 12:17:46.968 [scheduling-1] INFO  Sending data from 2022-09-14 to 2022-09-14
2022-09-14 12:17:47.478 [scheduling-1] INFO  Sending data to https://console-openshift-
console.apps.ls-js.cp.myibm.com/snapshot?token=*****
2022-09-14 12:17:47.640 [scheduling-1] INFO  Saving last hub synchronization date = 2022-09-14
2022-09-14 12:17:47.662 [scheduling-1] INFO  Sender task finished
```

If you find these entries in the License Service logs, it confirms that the License Service has connected to the License Service Reporter successfully.

Validating License Service Reporter deployment

Learn how to check whether License Service Reporter is properly deployed to your cluster and if it collects the complete license usage data from your cluster.

Complete the following steps to validate that License Service Reporter is successfully deployed and aggregates data from the clusters that you configured.

1. To check whether all the licensing pods in the `ibm-common-services` namespace are running, run the following command.

   - **Linux®:** `oc get pods -n ibm-common-services | grep "licens"`
   - **Windows:** `oc get pods -n ibm-common-services | findstr "licens"`

   An example of a response with the list of running pods.

   ```
   ibm-license-service-reporter-instance-66cfdc4c4f-xzxj7   3/3     Running             0
   37m
   ibm-licensing-operator-78d659f477-7lrqk                  1/1     Running             0
   41m
   ibm-licensing-service-instance-54c48dff64-jkvcr          1/1     Running             0
   8m23s
   ```

2. To check whether License Service Reported is successfully installed, verify that you can access the user interface. For detailed instructions, see Accessing the License Service Reporter user interface.

3. To check whether License Service instance is correctly configured on your clusters, run the /datasources API.For more information, see Retrieving information about data sources.

Viewing license usage on the Licensing dashboard

License Service Reporter shows the highest license usage in the environment on the Licensing dashboard.

You can access the Licensing dashboard from the Administration panel.

- Log in to Administration panel as a Cluster Administrator or a Cloud Pak Administrator and select **Licensing** from the main navigation menu.
- Log in to Administration panel as a Cluster Administrator or a Cloud Pak Administrator. After you log in, access the main navigation menu and click Home to access this dashboard. On the **Licensing cluster details** summary card, click **View license usage report**.
- Log in to the OpenShift console of the cluster where you deployed License Service Reporter. Go to **Networking** > **Routes**. Find the **cp-console** route and click the link under **Location**. Add the following part after the route: `/license-service-reporter`. For example: `https://cp-console.example/license-service-reporter`.

Note:

- After you deploy and configure License Service Reports, it takes around 30 minutes for the dashboard to load the licensing data. During this period, you see the **No data** message on your dashboard.
- The transfer of licensing data is not immediate, and the new data might show on the dashboard with the delay of up to 24 hours.

License Service Reporter shows the highest license usage of the IBM Cloud Paks and IBM stand-alone containerized software over a specified reporting period. The names of the clusters that the licensing data is based on are listed on the dashboard.

- Top-level view
- Deep dive
- No data on the dashboard
- Retrieving data with API

Top-level view

The top-level view shows the number of license metrics that is used by IBM containerized software that is detected in your environment within the reporting period. The value is based on the analysis of the license peaks from each cluster and it is the highest license usage that is registered in the environment.

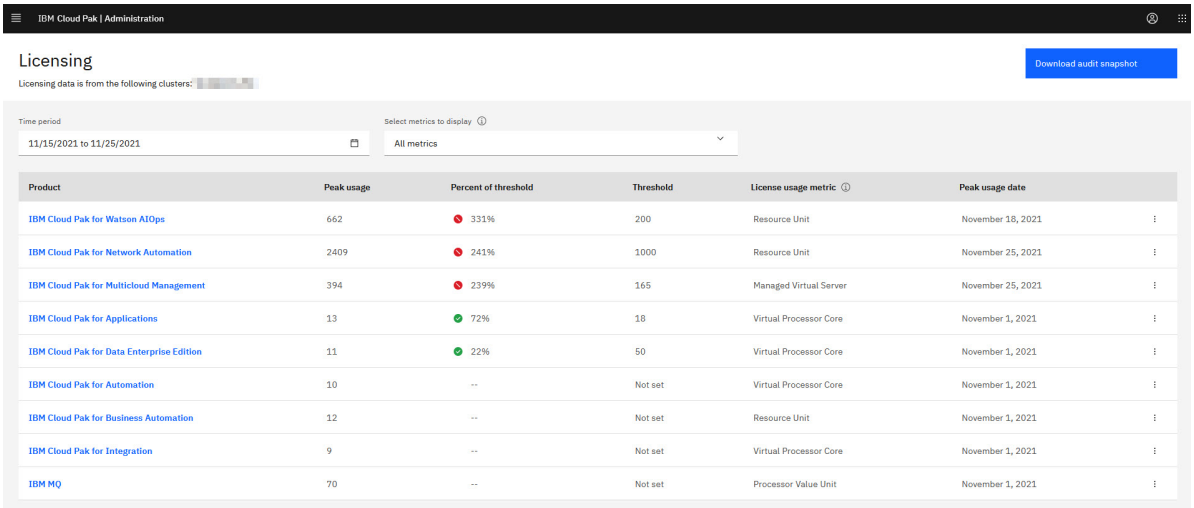


Figure. Top-level licensing dashboard view

Filtering options

- **Time range** - By default, the dashboard shows the current reporting period which is the current month. To change the time range, click the **Calendar** icon under **Time range** and select the period for which you want to view the highest license usage for your products. You can select a specific month, quarter, or year or set a custom period.
- **Select metrics to display** - By default, the dashboard shows information about all metrics. You can filter the view to see only the metric or metrics that you are interested in. To filter by metrics, click **All metrics** to expand the list and check the boxes next to the metrics that you want to display. The number of the metrics that are applied on the report is displayed in a grey box.

Columns

- **Product** - The name of the product that is deployed on the cluster.
- **Peak usage** - The highest license usage within the selected reporting period. License peak is calculated by adding the daily license usage of the product from all clusters. Then, the daily sums are compared. The highest sum is the value that is displayed as the peak usage. The status icon next to the peak usage value indicates whether the value is within the set threshold.
- **Percent of threshold** - The percent represents how the actual peak usage compares to the threshold that is set for the product.
- **Threshold** - The number of metric units that is specified for a product and can be compared with the peak usage.
- **License usage metric** - The name of the metric based on which the product is licensed.
- **Peak usage date** - The date when the license peak was recorded.

Sorting

You can sort the information on the dashboard by any column by clicking on the header.

By default, the products are sorted by the threshold status.

Setting the license usage threshold

You can set a license usage threshold for each product to see how it compares with your actual license usage. Threshold is a means of control that helps you better understand your actual license needs.

The threshold is the license usage limit that is expressed in a number of metric units. The threshold can, for example, base on the number of license metrics that you purchased, or be any other value that facilitates control in your environment. Setting the threshold is optional and its value can be modified at any time.

To set, reset or clear the threshold for a product, complete the following steps:

1. Click the overflow action menu next to the product and select **Threshold settings**.
2. Set the threshold.
    - To set or reset the threshold, specify the number of metric units.
    - To clear your current threshold, select **Do not set**.
3. Click **Save**.

After you set the threshold, the percent of threshold is automatically populated and the status icons that can help you control the limits appears before the percent. The icons indicate whether your current license usage is within the set threshold, according to the data collected from the clusters:

- The green tick indicates that the license usage for the product is below the set threshold.
- The yellow icon indicates that the license usage for the product is close or equal to the threshold.
- The red warning indicates that you exceed the set threshold. Consider what that can mean for your business.

The threshold can also be viewed and modified on the deep dive.

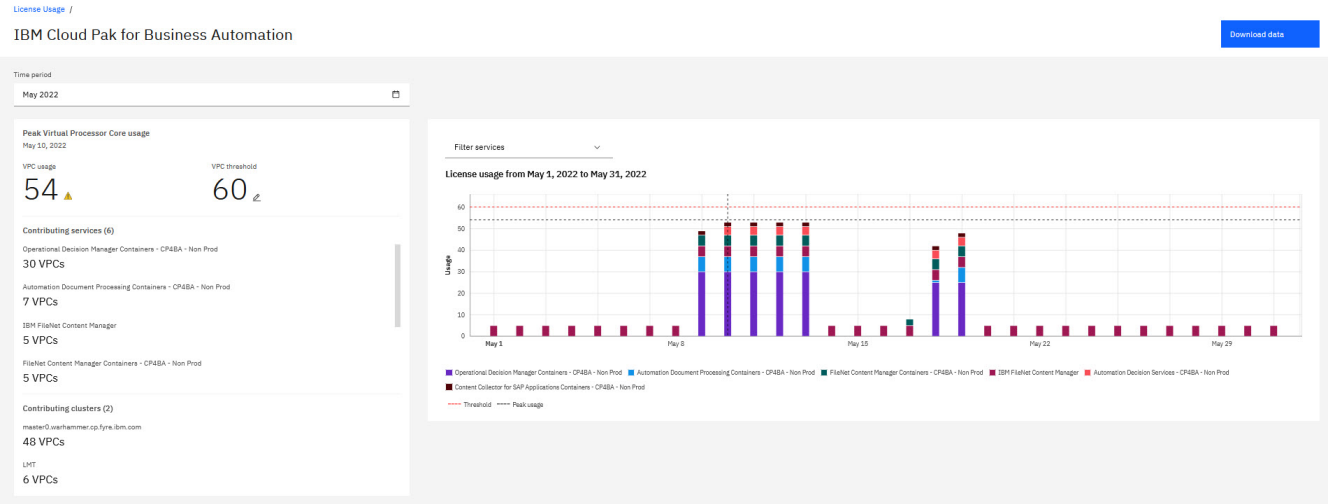
Generating audit snapshot

To generate an audit snapshot that is based on the selected criteria, click the **Download audit snapshot** button, and confirm by selecting **Download**.

When you download audit snapshot, you get a `.zip` package with a collection of audit snapshots for each cluster. Each audit snapshot is a compressed `.zip` package that includes a complete set of audit documents that certify your license usage per cluster. For more information about audit snapshot and its contents, see Audit snapshot.

Deep dive

The top-level view shows the highest license usage of IBM containerized software. To view the detailed information about the license usage of a specified product, click the name of the product and see the deep dive.



Time range

By default, the time range on the deep dive is inherited from the top-level view. You can customize the time range by clicking the **Calendar** icon.

License usage details

The deep dive shows the following information:

- **Peak license usage** - The date when the highest license usage for the product was registered within the set time range.
- **Contributing services** - The breakdown of the peak usage by bundled products that contribute to the overall license usage of the IBM Cloud Pak® within the set time range. When the bundled product is licensed with a different metric than the IBM Cloud Pak®, the original recorded metric of this bundled product is also displayed. Available for IBM Cloud Paks only.
- **Contributing clusters** The breakdown of the license usage peak by individual clusters within the set time range. Available for all products.
- **License usage** chart - The chart that presents the license usage of the product over time. For IBM Cloud Paks it additionally shows how the individual bundled products contributed to the highest license usage.

Click **Download data** to export the detailed information about the highest license usage for the product in the form of the .csv file. The file contains information about the contribution of the bundled products in the overall product license usage. In case of IBM stand-alone containerized products, the file contains only the information about the highest license usage of this product.

No data on the dashboard

After you deploy and configure License Service Reports, it takes around 30 minutes for the dashboard to load the licensing data. During this period, you see the **No data** message on your dashboard.

However, it might happen that the dashboard does not display any data after these 30 minutes, or at one point after you start using the dashboard. The following message is displayed instead:

No data. It looks like there is no data to display for selected period and metrics. For more information, see documentation.

The message can indicate that License Service Reporter did not receive any data for the selected metric in the selected reporting period. If you know of the products that should be reported, make sure that you configured License Service and enabled data feeds for the proper clusters. For more information, see Configuring data sources.

Retrieving the highest license usage for your multicluster environment with API

You can retrieve the highest license usage for your multicluster environment from License Service Reported with one of the dedicated APIs. For more information, see Retrieving license usage data for multiple clusters from License Service Reporter.

Tracking license usage in a multicluster environment

If you want to track the license usage of your product in a multicluster environment, you can manually add the values for each cluster to create a consolidated report.

Consolidating license usage data for tracking purposes

Note: Starting from IBM Cloud Pak® foundational services version 3.5.x, you can use License Service Reporter to track license usage in your environment in an automatized way. Deploy and configure License Service Reporter, and monitor the usage on a dedicated dashboard. For more information, see Tracking license usage in multicluster environment with License Service Reporter.

For smaller environments, or if you do not have resources for deploying and configuring License Service Reporter, you can use the following manual procedure.

Complete the following actions to create a consolidated report.

1. Get the **`products_daily_<reported_period>_<cluster hostname>.csv`** report that is a part of an audit snapshot for each cluster in your environment. For more information, see: Retrieving license metric usage and IBM License Service API schema.
2. Add the **`metricQuantity`** value of a product from a specific date for every cluster.
3. Record the sum for the date.
4. Repeat the preceding steps for each date within the requested period.
5. Find the highest cumulative value. This value is the highest license usage of the product within the reported period.

Example

Example 1. License usage of IBM Cloud Pak® A on Cluster 1 as it is reported in **`cluster1_products-daily-2020-03-01-2020-03-05_<cluster1 hostname>.csv`**.

Table. IBM Cloud Pak® A License usage on cluster 1

name	metricQuantityDate	metricQuantity	metricName
IBM Cloud Pak A	3/1/2020	5	VIRTUAL_PROCESSOR_CORE
IBM Cloud Pak A	3/2/2020	30	VIRTUAL_PROCESSOR_CORE
IBM Cloud Pak A	3/3/2020	30	VIRTUAL_PROCESSOR_CORE
IBM Cloud Pak A	3/4/2020	29	VIRTUAL_PROCESSOR_CORE
IBM Cloud Pak A	3/5/2020	29	VIRTUAL_PROCESSOR_CORE

Example 2. License usage of IBM Cloud Pak® A on Cluster 2 as it is reported in **`cluster2_products-daily-2020-03-01-2020-03-05_<cluster2 hostname>.csv`**.

Table. IBM Cloud Pak® A License usage on cluster 2

name	metricQuantityDate	metricQuantity	metricName
IBM Cloud Pak A	3/1/2020	5	VIRTUAL_PROCESSOR_CORE
IBM Cloud Pak A	3/2/2020	10	VIRTUAL_PROCESSOR_CORE
IBM Cloud Pak A	3/3/2020	10	VIRTUAL_PROCESSOR_CORE
IBM Cloud Pak A	3/4/2020	9	VIRTUAL_PROCESSOR_CORE
IBM Cloud Pak A	3/5/2020	9	VIRTUAL_PROCESSOR_CORE

Example 3. An example of how to prepare a table to find the highest license usage in the environment where IBM Cloud Pak A is deployed on Cluster 1 and Cluster 2.

Table. IBM Cloud Pak® A License usage on cluster 1 and 2

name	metricQuantity Date	metricQuantity on Cluster 1	Metric Quantity on Cluster 2	Total license usage	MetricName
IBM Cloud Pak A	3/1/2020	5	5	10	VIRTUAL_PROCESSOR_CORE
IBM Cloud Pak A	**3/2/2020**	**30**	**10**	**40**	**VIRTUAL_PROCESSOR_CORE**
IBM Cloud Pak A	3/3/2020	30	10	40	VIRTUAL_PROCESSOR_CORE
IBM Cloud Pak A	3/4/2020	29	9	38	VIRTUAL_PROCESSOR_CORE
IBM Cloud Pak A	3/5/2020	29	9	38	VIRTUAL_PROCESSOR_CORE

Passport Advantage part numbers

You might need to download a package from Passport Advantage® to install some products and extensions for IBM Cloud Pak® for AIOps. This requirement can be due to your entitlement or based on how you choose to install the product or extension.

Use these tables to find the files for IBM Cloud Pak® for AIOps that are included in Passport Advantage® and the corresponding part numbers.

- Downloading packages
- Quick start guide
- IBM Cloud Pak for AIOps Core packages
- IBM Cloud Pak for AIOps Supporting product packages
  - Infrastructure Automation packages
  - IBM Cloud Pak for AIOps Topology Manager product packages
  - IBM Cloud Pak for AIOps Metrics Manager product packages
  - IBM Cloud Pak for AIOps Extension eAssemblies

Downloading packages

Complete the following procedure to download your packages:

1. Go to the Passport Advantage Online tab at Passport Advantage and log in with your IBM ID.
2. Find your part number.
3. Search for your files by using the part numbers that are listed in the tables.
4. Download the files to a directory on your computer.

Core component eAssemblies

The following table lists the available multiplatform eAssembly number for IBM Cloud Pak for AIOps.

Table 1. Software eAssembly part numbers for base components

eAssembly #	Description
G09CZEN	IBM Cloud Pak for AIOps 4.2.0 Installation Packages eAssembly

Quick start guide

The following table lists the quick start guide eImage number for IBM Cloud Pak for AIOps.

Table 1. Software eImage part number for quick start guide

eImage #	Description
M0FL7EN	IBM Cloud Pak for AIOps 4.2.0 Quick Start Guide

Supporting product eAssemblies

The following tables list the available eAssembly numbers for IBM Cloud Pak® for AIOps supporting products.

Table 2. Software eAssembly part numbers for supporting products

eAssembly #	Description
G09D0ML	IBM Robotic Process Automation for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09DBML	IBM Process Mining for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09CGML	IBM Netcool Operations Insight V1.6.10 Quick Start Guide for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09CFML	IBM Netcool Operations Insight v1.6.10 Cloud Paks for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09CCML	IBM Tivoli Netcool/OMNIbus v8.1 for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09D6EN	IBM Tivoli Netcool Impact v7.1 for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09DFEN	IBM Operations Analytics Log Analysis v1.3.7 for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09DDML	IBM Operations Analytics Log Analysis Service Desk Extension v1.1 for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09DCML	IBM Operations Analytics Advanced Insights v1.3.6 for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09CHML	Jazz for Service Management 1.1.3.19 for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09C9ML	IBM Db2 Server Edition v11.5.8 from Netcool Operations Insight V1.6.8 for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09CNML	IBM Tivoli Monitoring for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09CMML	IBM Tivoli Monitoring Agents for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09CVML	IBM Tivoli Netcool/OMNIbus Probes for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09CKML	IBM Tivoli Netcool/OMNIbus Probe Dependencies for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09CTML	IBM Tivoli Netcool/OMNIbus Event Forwarding V8.1 for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09CLML	IBM Tivoli Netcool/OMNIbus Data Exchange V8.1 for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09D9ML	Network Health Dashboard v4.2 for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09CBML	Device Dashboard v1.1.0.2 for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09D1ML	IBM Cognos Analytic Server 11.1 for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09CSML	IBM Tivoli Monitoring V6.3.0.7 for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09CQML	IBM Tivoli Monitoring V6.3.0.2 QSG for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09CRML	IBM Tivoli Monitoring V6.3.0.7 Agent for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09C8ML	IBM DB2 Advanced Workgroup Server Edition 11.1 for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09C9ML	IBM DB2 Standard Edition 11.5.4 for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09CXML	IBM Tivoli System Automation For Multiplatform Base V3.2.0 for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09D5ML	Netcool System Service Monitor Component V4.0.1 for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09CJML	Jazz for Service Management V1.1.2.0 for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09CJML	ITCAM Transactions V7.4.0.1 Response Time and Internet SerVice Monitoring for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09C3ML	ITCAM Transactions V7.4.0.1 Transaction Tracking for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09C6ML	IBM Rational 8.6 for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09C1ML	ITCAM for Microsoft Applications Version 6.3.1 for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09C0ML	ITCAM for Applications V7.2.2.2 for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09DGML	IBM Tivoli Monitoring for Virtual Environments V7.2.0.3 QSG for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09BZML	ITCAM for Application Diagnostics V7.1.0.4: Managing Server Component for IBM Cloud Pak for AIOps 4.2.0 eAssembly

Infrastructure Automation packages

The following table lists the eAssembly number for Infrastructure Automation - Infrastructure management.

Table. Software eAssembly part numbers for Infrastructure Automation

eAssembly #	Description
G09CYEN	IBM Cloud Pak for AIOps 4.2.0 Infrastructure Automation eAssembly

The following table lists the eImage numbers for Infrastructure Automation - Infrastructure management.

Table. Software eImage part number for Infrastructure Automation

eImage #	Description
M0FL8EN	IBM Cloud Pak for AIOps 4.2.0 Infrastructure automation for VMware vSphere with PVSCSI controller
M0FL9EN	IBM Cloud Pak for AIOps 4.2.0 Infrastructure automation for Red Hat OpenStack Platform
M0FLBEN	IBM Cloud Pak for AIOps 4.2.0 Infrastructure automation for Amazon EC2
M0FLCEN	IBM Cloud Pak for AIOps 4.2.0 Infrastructure automation for Red Hat Virtualization
M0FLDEN	IBM Cloud Pak for AIOps 4.2.0 Infrastructure automation RPM updates

Topology Manager eAssemblies

Table 4. Software eAssembly part numbers for Topology Manager supporting products

eAssembly #	Description
G09D3ML	IBM Netcool Agile Service Manager v1.1 Base for IBM Cloud Pak for AIOps 4.2.0 eAssembly
G09D4ML	IBM Netcool Agile Service Manager v1.1 Observers for IBM Cloud Pak for AIOps 4.2.0 eAssembly

Metrics Manager eAssemblies

Table 5. Software eAssembly part numbers for Metrics Manager supporting products

eAssembly #	Description
G09D8EN	IBM Operations Analytics Predictive Insights v1.3.6 for IBM Cloud Pak for AIOps 4.2.0 eAssembly

Other supporting product and extension eAssemblies

The following table lists the available eAssembly numbers for IBM Cloud Pak® for AIOps extensions.

Table 6. Software eAssembly part numbers for extensions

eAssembly #	Description
G09C7EN	IBM Cloud Pak for AIOps V4.2.0 Extensions Quick Start Guide eAssembly
G09CPML	IBM Tivoli Network Manager IP Edition 4.2 for IBM Cloud Pak for AIOps V4.2.0 Extensions eAssembly
G09CWML	IBM Tivoli Monitoring V6.3.0.2 for Tivoli Network Manager IP Edition V4.2 for IBM Cloud Pak for AIOps V4.2.0 Extensions eAssembly
G09C4ML	IBM Netcool Configuration Manager v6.4.2.18 Base for IBM Cloud Pak for AIOps V4.2.0 Extensions eAssembly
G09C5ML	IBM Netcool Configuration Manager v6.4.2.18 IBM JRE Java (TM) V8.0.8.6 for IBM Cloud Pak for AIOps V4.2.0 Extensions eAssembly
G09CDML	IBM Netcool Configuration Manager v6.4 Driver 24 for IBM Cloud Pak for Watson AIOps 4.1.0 Extensioms eAssembly
G09D2ML	IBM Netcool Agile Service Manager v1.1 Application and Network Discovery for IBM Cloud Pak for AIOps V4.2.0 Extensions eAssembly

Installing IBM Cloud Pak for AIOps

You can choose to complete a standard production environment installation of IBM Cloud Pak® for AIOps for a starter or production deployment, or a streamlined small starter installation for a non-production proof-of-concept or trial deployment.

Starter installation

A starter installation provides a concise installation of IBM Cloud Pak for AIOps for deployments that do not require a sustained workload. Use a starter installation only for proof-of-concept or experimentation environments.

- Online starter installation of IBM Cloud Pak for AIOps

Note: If you require an air-gapped (offline) deployment, you must complete a production installation.

Production installation

Production installations are intended for real world use cases and typically require more system and storage resources.

- Installing IBM Cloud Pak for AIOps (Production installation)

Alternatively, you can use GitOps to install IBM Cloud Pak for AIOps. For more information, see Deploy IBM Cloud Pak for AIOps using GitOps.

Online starter installation of IBM Cloud Pak for AIOps (CLI)

If you are a trial user of IBM Cloud Pak® for AIOps, or you want a proof-of-concept deployment that does not require a sustained workload, consider a starter installation to get a smaller, nonproduction deployment up and running quickly.

If you are installing on Azure Red Hat OpenShift (ARO), Google Cloud Platform (GCP), Red Hat® OpenShift® on IBM Cloud® service (ROKS), or Red Hat® OpenShift® Service on AWS (ROSA) then review the information in Installing on cloud platforms.

If you require a production-grade deployment, or an offline air-gapped deployment, see Installing IBM Cloud Pak for AIOps.

Before you begin

- Review the Planning section and ensure that your cluster meets the system requirements for a starter deployment.
- Online installations of IBM Cloud Pak for AIOps can be run entirely as a non-root user, and do not require that user to have sudo access.
- Ensure that you are logged in to your Red Hat OpenShift cluster with `oc login` for any steps that use the Red Hat OpenShift command-line interface (CLI).
- If you require details about the permissions that the IBM Cloud Pak for AIOps operators need, see Permissions (IBM Cloud Pak for AIOps).
- Red Hat® OpenShift® Container Platform requires a user with `cluster-admin` privileges for the following operations:
  - Create a custom project
  - Create the catalog source
  - Create an OperatorGroup

Limitations

- After installation, you cannot increase the size of your IBM Cloud Pak for AIOps deployment from a starter deployment to a larger production deployment.
- High availability is not supported for starter deployments.
- This starter installation uses the `OwnNamespace` installation mode, which is the recommended installation mode for IBM Cloud Pak for AIOps, to provide better resilience and performance. For more information, see Operator installation mode.

1. Install and configure Red Hat OpenShift Container Platform

IBM Cloud Pak for AIOps requires Red Hat OpenShift to be installed and running. You must have administrative access to your Red Hat OpenShift cluster.

For more information about the supported versions of Red Hat OpenShift, see Supported Red Hat OpenShift Container Platform versions. The hardware architecture that you install IBM Cloud Pak for AIOps on must be AMD64.

1. Install Red Hat OpenShift by using the instructions in the Red Hat OpenShift documentation.
2. Install the Red Hat OpenShift (**oc**) command-line interface (CLI) on your cluster's boot node and run **oc login**. For more information, see Getting started with the Red Hat OpenShift CLI.
3. Ensure that the clocks on your Red Hat OpenShift cluster are synchronized. Each Red Hat OpenShift node in the cluster must have access to an NTP server. Red Hat OpenShift nodes use NTP to synchronize their clocks. IBM Cloud Pak for AIOps runs on Red Hat OpenShift and also has this requirement. Discrepancies between the clocks on the Red Hat OpenShift nodes can cause IBM Cloud Pak for AIOps to experience operational issues. See the Red Hat OpenShift documentation for information about how to use a **MachineConfig** custom resource to configure chrony to connect to your NTP servers.
4. Optionally configure a custom certificate for IBM Cloud Pak for AIOps to use. You can use either of the following methods:
   - Configure a custom certificate for the Red Hat OpenShift cluster. Follow the instructions in the Red Hat OpenShift documentation Replacing the default ingress certificate. Then, deploy the signing CA certificate into the cluster by following the instructions in the Red Hat OpenShift documentation Replacing the CA Bundle certificate.
   - If you would like to use a custom certificate for IBM Cloud Pak for AIOps only, then after installation is complete follow the instructions in Using a custom certificate.

2. Configure storage

Configure your own storage for use with IBM Cloud Pak for AIOps. For more information, see Storage.

3. Create a custom project (namespace)

Create a project (namespace) called **cp4aiops** for your IBM Cloud Pak for AIOps deployment, by running the following command:

```
oc create namespace cp4aiops
```

4. Create an OperatorGroup in your custom project (namespace)

Create the Operator group by running the following command:

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: cp4aiops-operator-group
  namespace: cp4aiops
spec:
  targetNamespaces:
    - cp4aiops
EOF
```

5. Create the entitlement key pull secret

1. Log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.
2. In the *Entitlement keys* section, select **Copy** to copy your entitlement key to the clipboard.
3. From the Red Hat OpenShift CLI, run the following command:

```
oc create secret docker-registry ibm-entitlement-key \
    --docker-username=cp \
    --docker-password=<entitlement-key> \
    --docker-server=cp.icr.io \
    --namespace=cp4aiops
```

Where **<entitlement-key>** is the entitlement key that you copied in the previous step.

6. Configure usage data collection

To help the development of IBM Cloud Pak for AIOps, daily aggregated usage data is collected to analyse how IBM Cloud Pak for AIOps is used. The usage data is collected by the **cp4waiops-metricsprocessor** pod, and is sent to and stored in IBM controlled GDPR-compliant systems. The collection of usage data is enabled by default, but can be disabled. For transparency, the **cp4waiops-metricsprocessor** pod's logs contain all the information that is collected. The usage data that is collected is all numeric, and does not include email addresses, passwords, or specific details. Only the following data is collected:

- Current number of applications
- Current number of alerts (all severities aggregated)
- Current number of incidents (all priorities aggregated)
- Current number of policies (includes predefined and user created)
- Current number of runbooks run since installation
- Current number of connectors of each type (For example ServiceNow, Instana, Falcon Logscale)

Configuring the collection of usage data

If you do not want to disable the collection of usage data, run the following steps.

1. Set environment variables.

```
export CUSTOMER_NAME=<your company name>
export CUSTOMER_ICN=<your IBM Customer Number>
export CUSTOMER_ENVIRONMENT=<Set to `trial` or `poc`>
export PROJECT_CP4AIOPS=cp4aiops
```

2. Configure the usage data with your customer details.

```
oc create secret generic aiops-metrics-processor -n ${PROJECT_CP4AIOPS} --from-
literal=customerName=${CUSTOMER_NAME} --from-literal=customerICN=${CUSTOMER_ICN} --from-
literal=environment=${CUSTOMER_ENVIRONMENT}
```

3. If you have a firewall enabled, ensure that outbound traffic to https://api.segment.io is allowed.

Important: Usage data without your customer details is still collected even if you do not create this secret. If you do not want any usage data collected, then you must run the command given in *Disabling the collection of usage data*.

Disabling the collection of usage data

If you want to disable the collection of usage data, run the following commands.

1. Set environment variables.

```
export CUSTOMER_NAME=<your company name>
export CUSTOMER_ICN=<your IBM Customer Number>
export CUSTOMER_ENVIRONMENT=<Set to `trial`, `poc`, or `production`>
export PROJECT_CP4AIOPS=cp4aiops
```

2. Disable usage data collection

```
oc create secret generic aiops-metrics-processor -n ${PROJECT_CP4AIOPS} --from-
literal=customerName=${CUSTOMER_NAME} --from-literal=customerICN=${CUSTOMER_ICN} --from-
literal=environment=${CUSTOMER_ENVIRONMENT} --from-literal=enableCollection=false
```

Note: You can update your usage data collection preferences after installation. For more information, see Updating usage data collection preferences.

7. Create the catalog source

Add the IBM Cloud Pak for AIOps catalog source to your Red Hat OpenShift cluster.

After installation, the **ibm-operator-catalog** CatalogSource object determines whether the upgrade of your IBM Cloud Pak for AIOps deployment is initiated automatically when a new patch becomes available. The **ibm-operator-catalog** CatalogSource object can be configured to automatically poll for and retrieve a newer catalog by enabling the polling attribute, **spec.updateStrategy.registryPoll**. If a newer catalog for a patch is found and retrieved, then an automatic upgrade of your IBM Cloud Pak for AIOps deployment is initiated. For more information, see Controlling upgrade.

Note: **ibm-operator-catalog** also contains the catalogs for other IBM Cloud Paks®. If multiple IBM Cloud Paks are installed on your cluster, then the polling attribute is configured for all of them.

If you want to enable the automatic initiation of patch upgrades, run the following command to create the **ibm-operator-catalog** CatalogSource object with polling enabled.

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-operator-catalog
  publisher: IBM Content
  sourceType: grpc
  image: icr.io/cpopen/ibm-operator-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
EOF
```

If you want to disable the automatic initiation of patch upgrades, then use the following steps.

1. Run the following command to create the **ibm-operator-catalog** CatalogSource object without polling enabled.

   ```
   cat << EOF | oc apply -f -
   apiVersion: operators.coreos.com/v1alpha1
   kind: CatalogSource
   metadata:
     name: ibm-operator-catalog
     namespace: openshift-marketplace
   spec:
     displayName: ibm-operator-catalog
     publisher: IBM Content
     sourceType: grpc
     image: icr.io/cpopen/ibm-operator-catalog:latest
   EOF
   ```

2. Update the **ibm-operator-catalog** CatalogSource to fix it to always use the current image digest, instead of **icr.io/cpopen/ibm-operator-catalog:latest**. This ensures that the **ibm-operator-catalog** CatalogSource pods do not pull the latest image if a node reload or other issue causes them to restart. Run the following commands:

   ```
   IMGDIGEST=`oc get pods -n openshift-marketplace -l=olm.catalogSource=ibm-operator-catalog
   --no-headers -o=jsonpath="{.items[0].status.containerStatuses[0].imageID}" -n openshift-
   marketplace`
   CATALOGIMG=`oc get catalogsource ibm-operator-catalog -n openshift-marketplace -o
   jsonpath='{.spec.image}'`
   if [ "${CATALOGIMG}" = "icr.io/cpopen/ibm-operator-catalog:latest" ]; then
     oc patch catalogsource ibm-operator-catalog -n openshift-marketplace --type=json -p "[{
   "op": "replace", "path": "/spec/image", "value": "\"$IMGDIGEST\"" }]"
   fi
   ```

You can disable or re-enable automatic patch upgrade after installation if you change your mind. For more information, see Configuring automatic patch upgrades.

8. Verify cluster readiness

Run the following command to switch to your IBM Cloud Pak for AIOps project, 'cp4aiops'.

```
oc project cp4aiops
```

Run the prerequisite checker script to verify whether your Red Hat OpenShift cluster is correctly set up for an IBM Cloud Pak for AIOps installation. For more information about the script, and how to download and run it, see github.com/IBM.

Example output:

```
# ./prereq.sh
Using project "cp4aiops" on server "https://myserver.mycluster.mydomain:6443".

Starting IBM Cloud Pak for AIOps prerequisite checker v4.2...

[INFO] ================================Openshift Container Platform Version
Check================================
[INFO] Checking OCP Version. Compatible Versions of OCP is v4.12.x
[INFO] OCP version 4.12.22 is compaitble
[INFO] ================================Openshift Container Platform Version
Check================================

[INFO] ================================Entitlement Pull
Secret================================
[INFO] Checking whether the Entitlement secret or Global pull secret is configured correctly.
[INFO] Checking if the job 'cp4aiops-entitlement-key-test-job' already exists.
[INFO] The job with name 'cp4aiops-entitlement-key-test-job' was not found, so moving ahead
and creating it.
[INFO] Creating the job 'cp4aiops-entitlement-key-test-job'
job.batch/cp4aiops-entitlement-key-test-job created
[INFO] Verifying if the job 'cp4aiops-entitlement-key-test-job' completed successfully..
[INFO] SUCCESS! Entitlement secret is configured correctly.
job.batch "cp4aiops-entitlement-key-test-job" deleted
[INFO] ================================Entitlement Pull
Secret================================

[INFO] ================================Storage Provider================================
[INFO] Checking storage providers
[INFO] No IBM Storage Fusion Found... Skipping configuration check.

[INFO] No Portworx StorageClusters found with "Running" or "Online" status. Skipping
configuration check for Portworx.
[INFO] Openshift Data Foundation found.
[INFO] No IBM Cloud Storage found... Skipping configuration check for IBM Cloud Storage Check.

Checking Openshift Data Foundation Configuration...
Verifying if Red Hat Openshift Data Foundation pods are in "Running" or "Completed" status
[INFO] Pods in openshift-storage project are "Running" or "Completed"
[INFO] ocs-storagecluster-ceph-rbd exists.
[INFO] ocs-storagecluster-cephfs exists.
[INFO] No warnings or failures found when checking for Storage Providers.
[INFO] ================================Storage Provider================================

[INFO] ================================Small or Large Profile Install
Resources================================
[INFO] Checking for cluster resources

[INFO] ================================Resource
Summary=========================================================
[INFO]                                        Nodes    |      vCPU       |     Memory(GB)
[INFO] Small profile(available/required)   [  14 / 3 ]   [  123 / 62 ]       [  159 / 140 ]
[INFO] Large profile(available/required)   [  14 / 10 ]  [  123 / 162 ]      [  159 / 372 ]
[INFO] ================================Resource
Summary=========================================================
[INFO] Cluster currently has resources available to create a small profile of Cloud Pak for
AIOps

[INFO] ================================Small or Large Profile Install
Resources================================


[INFO] ================================Prerequisite Checker Tool
Summary================================
      [  PASS  ] Openshift Container Platform Version Check
      [  PASS  ] Entitlement Pull Secret
      [  PASS  ] Storage Provider
      [  PASS  ] Small or Large Profile Install Resources
```

```
[INFO] ================================Prerequisite Checker Tool
Summary================================
```

9. Install the IBM Cloud Pak for AIOps operator

For more information about the operators which are installed with IBM Cloud Pak for AIOps, see Operator Details.

Run the following command:

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-aiops-orchestrator
  namespace: cp4aiops
spec:
  channel: v4.2
  installPlanApproval: Automatic
  name: ibm-aiops-orchestrator
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
EOF
```

Warning: installPlanApproval must not be changed to *Manual*. Manual approval, which requires the manual review and approval of the generated InstallPlans, is not supported. Incorrect timing or ordering of manual approvals of InstallPlans can result in a failed installation.

After a few minutes, the IBM Cloud Pak for AIOps operator is installed. Verify that the all of the components have a state of **Succeeded** by running the following command:

```
oc get csv -n cp4aiops | egrep "ibm-aiops-orchestrator|ibm-common-service"
```

Example output:

```
$ oc get csv -n cp4aiops | egrep "ibm-aiops-orchestrator|ibm-common-service"
ibm-aiops-orchestrator.v4.2.0         IBM Cloud Pak for AIOps               4.2.0   Succeeded
ibm-common-service-operator.v3.23.7   IBM Cloud Pak foundational services   3.23.7  Succeeded
```

10. Install IBM Cloud Pak for AIOps

1. Run the following command to create an instance of the IBM Cloud Pak for AIOps custom resource called **ibm-cp-aiops**.

   Warning: The pakModules **aiopsFoundation**, **applicationManager**, and **aiManager** must be enabled as in the following YAML. Do not change these values to false.

   ```
   cat << EOF | oc apply -f -
   apiVersion: orchestrator.aiops.ibm.com/v1alpha1
   kind: Installation
   metadata:
     name: ibm-cp-aiops
     namespace: cp4aiops
   spec:
     imagePullSecret: ibm-entitlement-key
     license:
       accept: <license_acceptance>
     pakModules:
     - name: aiopsFoundation
       enabled: true
     - name: applicationManager
       enabled: true
     - name: aiManager
       enabled: true
     - name: connection
   ```

```
    enabled: false
  size: small
  storageClass: <storage_class_name>
  storageClassLargeBlock: <large_block_storage_class_name>
  topologyModel: application
EOF
```

Where

- **<license_acceptance>** - is set to *true* to agree to the license terms.
- **<storage_class_name>** and **<large_block_storage_class_name>** are the storage classes that you want to use, as detailed in the following table. For more information about storage, see Storage.

Note: To confirm that you have the storage classes for your chosen storage provider as shown in the table, run **oc get sc**.

Table 1. Storage provider classes

Storage provider	storage_class_name	large_block_storage_class_name
IBM Cloud®	ibmc-file-gold-gid	ibmc-block-gold
Red Hat® OpenShift® Data Foundation	ocs-storagecluster-cephfs	ocs-storagecluster-ceph-rbd
IBM Storage Fusion	ibm-spectrum-scale-sc	ibm-spectrum-scale-sc
IBM Spectrum Scale Container Native	ibm-spectrum-scale-sc	ibm-spectrum-scale-sc
Portworx	portworx-fs	portworx-aiops

2. Verify your installation.

   Run the following command to check that the **PHASE** of your installation is **Updating**.

   ```
   oc get installations.orchestrator.aiops.ibm.com -n cp4aiops
   ```

   Example output:

   ```
   NAME            PHASE      LICENSE    STORAGECLASS    STORAGECLASSLARGEBLOCK    AGE
   ibm-cp-aiops    Updating   Accepted   rook-cephfs     rook-ceph-block           3m
   ```

   It takes around 60-90 minutes for the installation to complete (subject to the speed with which images can be pulled). When installation is complete and successful, the **PHASE** of your installation changes to **Running**. If your installation phase does not change to **Running**, then use the following command to find out which components are not ready:

   ```
   oc get installation.orchestrator.aiops.ibm.com -o yaml -n cp4aiops | grep 'Not Ready'
   ```

   Example output:

   ```
   lifecycleservice: Not Ready
   zenservice: Not Ready
   ```

   To see details about why a component is **Not Ready** run the following command, where **<component>** is the component that is not ready, for example **zenservice**.

   ```
   oc get <component> -o yaml -n cp4aiops
   ```

   (Optional) You can also download and run a status checker script to see information about the status of your deployment. For more information about how to download and run the script, see github.com/IBM.

If the installation fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade and Known Issues to help you identify any installation problems.

11. Log in to the IBM Cloud Pak Automation console

1. Find the password for the **admin** username by running the following command:

   ```
   oc -n ibm-common-services get secret platform-auth-idp-credentials -o
   jsonpath='{.data.admin_password}' | base64 -d
   ```

2. Find the URL to access the IBM Cloud Pak Automation console with the following command.

   ```
   oc get route -n cp4aiops cpd -o jsonpath='{.spec.host}'
   ```

   The following output is a sample output:

   ```
   cpd-cp4aiops.apps.mycluster.mydomain
   ```

   Based on the sample output, your console URL would be **https://cpd-cp4aiops.apps.mycluster.mydomain**

3. Enter the URL in your browser to open the IBM Cloud Pak Automation console and log in with a username of **admin** and the password that you found in the previous step.

What to do next

- Define integrations and applications with Defining.
- If you have an existing on-premises IBM Tivoli Netcool/OMNIbus deployment, then you can connect IBM Cloud Pak for AIOps to it with the Netcool connector. For more information, see Creating IBM Tivoli Netcool/OMNIbus connections.
- If you have an existing on-premises IBM Tivoli Netcool/Impact deployment, then you can connect IBM Cloud Pak for AIOps to it with the IBM Tivoli Netcool/Impact connector. For more information, see Creating IBM Tivoli Netcool/Impact connections

Installing IBM Cloud Pak for AIOps (Production installation)

Depending on whether your OpenShift cluster is connected to the internet, choose either an online or an offline installation.

If you are installing on Azure Red Hat OpenShift (ARO), Google Cloud Platform (GCP), Red Hat® OpenShift® on IBM Cloud® service (ROKS), or Red Hat® OpenShift® Service on AWS (ROSA) then review the information in Installing on cloud platforms.

Online installation

If your cluster is connected to the internet, then you can install IBM Cloud Pak® for AIOps using the Red Hat® OpenShift® Container Platform console or the command line interface (CLI). Choose one of the following topics.

- Online production installation of IBM Cloud Pak for AIOps (console method)
- Online production installation of IBM Cloud Pak for AIOps (CLI method)

Offline installation

If your Red Hat OpenShift cluster is not connected to the internet, then you can perform an airgapped (offline) installation.

- Installing IBM Cloud Pak for AIOps in an air-gapped environment (offline)

Online production installation of IBM Cloud Pak for AIOps (console method)

If your cluster is connected to the internet, you can complete a production installation of IBM Cloud Pak® for AIOps with the Red Hat® OpenShift® Container Platform console.

Before you begin

- Review the Planning section.
- Online installations of IBM Cloud Pak for AIOps can be run entirely as a non-root user, and do not require that user to have sudo access.

- Some steps must still be run with the command line. Ensure that you are logged in to your Red Hat OpenShift cluster with `oc login` for any steps that use the Red Hat OpenShift command-line interface (CLI).
- The display names of some Red Hat OpenShift console components, such as window titles and push buttons, vary between Red Hat OpenShift versions. The following instructions are based on Red Hat OpenShift version 4.12 console components.
- If you require details about the permissions that the IBM Cloud Pak for AIOps operators need, see Permissions (IBM Cloud Pak for AIOps).
- Red Hat OpenShift requires a user with `cluster-admin` privileges for the following operations:
  - Create a custom project
  - Create the catalog source

Installation procedure

Follow these steps to install IBM Cloud Pak for AIOps.

1. Install and configure Red Hat OpenShift
2. Configure storage
3. Create a custom project (namespace)
4. Create the entitlement key secret
5. Configure usage data collection
6. Create the catalog source
7. Verify cluster readiness
8. Install the operator
9. Create the custom resource
10. Create an EgressFirewall to restrict outgoing traffic
11. Access the IBM Cloud Pak Automation console

1. Install and configure Red Hat OpenShift

IBM Cloud Pak for AIOps requires Red Hat OpenShift to be installed and running. You must have administrative access to your Red Hat OpenShift cluster.

For more information about supported versions of Red Hat OpenShift, see Supported Red Hat OpenShift Container Platform versions. The hardware architecture that you install IBM Cloud Pak for AIOps on must be AMD64.

1. Install Red Hat OpenShift by using the instructions in the Red Hat OpenShift documentation.

2. Install the Red Hat OpenShift command line interface (`oc`) on your cluster's boot node and run `oc login`. For more information, see the instructions in Getting started with the Red Hat OpenShift CLI.

3. Ensure that the clocks on your Red Hat OpenShift cluster are synchronized. Each Red Hat OpenShift node in the cluster must have access to an NTP server. Red Hat OpenShift nodes use NTP to synchronize their clocks. IBM Cloud Pak for AIOps runs on Red Hat OpenShift and also has this requirement. Discrepancies between the clocks on the Red Hat OpenShift nodes can cause IBM Cloud Pak for AIOps to experience operational issues. See the Red Hat OpenShift documentation for information about how to use a `MachineConfig` custom resource to configure chrony to connect to your NTP servers.

4. Optionally configure a custom certificate for IBM Cloud Pak for AIOps to use. You can use either of the following methods:

   - Configure a custom certificate for the Red Hat OpenShift cluster. Follow the instructions in the Red Hat OpenShift documentation Replacing the default ingress certificate. Then, deploy the signing CA certificate into the cluster by following the instructions in the Red Hat OpenShift documentation Replacing the CA Bundle certificate.
   - If you would like to use a custom certificate for IBM Cloud Pak for AIOps only, then after installation is complete follow the instructions in Using a custom certificate.

2. Configure storage

The storage configuration must satisfy your sizing requirements. For more information about the storage classes that are needed for installing IBM Cloud Pak for AIOps, see Storage.

3. Create a custom project (namespace)

Create a project (namespace) to deploy IBM Cloud Pak for AIOps into.

A project is a Kubernetes namespace. You must create a custom project (namespace) and not use the **`default`**, **`kube-system`**, **`kube-public`**, **`openshift-node`**, **`openshift-infra`**, or **`openshift`** projects (namespaces). This is because IBM Cloud Pak for AIOps uses Security Context Constraints (SCC), and SCCs cannot be assigned to pods created in one of the default Red Hat OpenShift projects (namespaces).

1. From your Red Hat OpenShift console, click **Home** > **Projects**.
2. Select **Create Project**, specify the **Name** of the project that you want to create, for example *cp4aiops* and click **Create**.

4. Create the entitlement key secret

Complete the following steps to create a docker-registry secret to enable your deployment to pull the IBM Cloud Pak for AIOps images from the IBM® Entitled Registry.

1. Obtain the entitlement key that is assigned to your IBMid. Log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.
2. In the *Active entitlement keys* section, select **Copy** to copy the entitlement key to the clipboard.
3. From your Red Hat OpenShift console, click **Workloads** > **Secrets**.
4. From the **Project** menu, select the project that you created earlier in Create a custom project (namespace).
5. Click the **Create** button, and select **Image Pull Secret** from the menu. The **Create image pull secret** form is displayed. Enter the following values and then click **Create**.
   - **Secret name**: `ibm-entitlement-key`
   - **Authentication type**: `Image registry credentials`
   - **Registry server address**: `cp.icr.io`
   - **Username**: `cp`
   - **Password**: use the entitlement key that you copied in step 2.

5. Configure usage data collection

To help the development of IBM Cloud Pak for AIOps, daily aggregated usage data is collected to analyse how IBM Cloud Pak for AIOps is used. The usage data is collected by the **`cp4waiops-metricsprocessor`** pod, and is sent to and stored in IBM controlled GDPR-compliant systems. The collection of usage data is enabled by default, but can be disabled. For transparency, the **`cp4waiops-metricsprocessor`** pod's logs contain all the information that is collected. The usage data that is collected is all numeric, and does not include email addresses, passwords, or specific details. Only the following data is collected:

- Current number of applications
- Current number of alerts (all severities aggregated)
- Current number of incidents (all priorities aggregated)
- Current number of policies (includes predefined and user created)
- Current number of runbooks run since installation
- Current number of connectors of each type (For example ServiceNow, Instana, Falcon Logscale)

Use the following steps to configure or disable usage data collection.

1. From your Red Hat OpenShift console, click **Workloads** > **Secrets**.
2. From the **Project** menu, select the project that you created earlier in Create a custom project (namespace).
3. Click the **Create** button, and select **Key/value secret** from the menu. The **Create key/value secret** form is displayed. Enter the following values and then click **Create**.
   - **Secret name**: `aiops-metrics-processor`
   - Add the following **Key**/**Value** pairs:

- **customerName**: your company name
- **customerICN**: your IBM Customer Number (ICN)
- **environment**: `trial` for testing, `poc` for proof of concept, or `production` for production environments.

- If you want to disable usage data collection, also add the following key/value pair: **enableCollection**: false

4. If you have a firewall enabled, ensure that outbound traffic to https://api.segment.io is allowed.

Important: Usage data without your customer details is still collected even if you do not create this secret. If you do not want any usage data collected, then you must create this secret with **enableCollection** set to false.

You can update your usage data collection preferences after installation. For more information, see Updating usage data collection preferences.

6. Create the catalog source

Add the IBM Cloud Pak for AIOps catalog source to your Red Hat OpenShift cluster.

After installation, the `ibm-operator-catalog` CatalogSource object determines whether the upgrade of your IBM Cloud Pak for AIOps deployment is initiated automatically when a new patch becomes available. The `ibm-operator-catalog` CatalogSource object can be configured to automatically poll for and retrieve a newer catalog by enabling the polling attribute, `spec.updateStrategy.registryPoll`. If a newer catalog for a patch is found and retrieved, then an automatic upgrade of your IBM Cloud Pak for AIOps deployment is initiated. For more information, see Controlling upgrade.

Note: `ibm-operator-catalog` also contains the catalogs for other IBM Cloud Paks®. If multiple IBM Cloud Paks are installed on your cluster, then the polling attribute is configured for all of them.

1. Log in to your Red Hat OpenShift cluster's console.

2. Add the `IBM Operators` CatalogSource.

   If you want to enable the automatic initiation of patch upgrades, click the plus icon in the upper right corner to open the **Import YAML** dialog box, paste in the following YAML, and then click **Create**.

   ```
   apiVersion: operators.coreos.com/v1alpha1
   kind: CatalogSource
   metadata:
     name: ibm-operator-catalog
     namespace: openshift-marketplace
   spec:
     displayName: ibm-operator-catalog
     publisher: IBM Content
     sourceType: grpc
     image: icr.io/cpopen/ibm-operator-catalog:latest
     updateStrategy:
       registryPoll:
         interval: 45m
   ```

   If you want to disable the automatic initiation of patch upgrades, then use the following steps that do not configure catalog polling, and which also fix the image to the current image digest only. This ensures that the latest image is not pulled if a node reload or other issue causes the `ibm-operator-catalog` CatalogSource pods to restart.

   1. Click the plus icon in the upper right to open the **Import YAML** dialog box, paste in the following YAML, and then click **Create**.

      ```
      apiVersion: operators.coreos.com/v1alpha1
      kind: CatalogSource
      metadata:
        name: ibm-operator-catalog
        namespace: openshift-marketplace
      spec:
        displayName: ibm-operator-catalog
        publisher: IBM Content
        sourceType: grpc
        image: icr.io/cpopen/ibm-operator-catalog:latest
      ```

   2. Find the current image digest.

      1. Go to **Home** > **Projects**, and select `openshift-marketplace`.

2. Go to **Workloads** > **Pods** (on the left menu), and then search for `ibm-operator-catalog`.
3. Click the returned `ibm-operator-catalog-<...>` pod.
4. Click **YAML** to switch to the YAML view.
5. Search for `imageID` in the YAML, and copy down the value of `spec.containerStatuses.imageID`. The value is in a format similar to the following example:

   ```
   icr.io/cpopen/ibm-operator-catalog@sha256:<...>
   ```

3. Go to **Administration** > **Cluster Settings**. Under **Configuration** > **OperatorHub** > **Sources**, scroll down and click `ibm-operator-catalog`.
4. Click **YAML** to switch to the YAML view.
5. Set the value of `spec.image` to the value of the current image digest that you found in step 2, instead of to `icr.io/cpopen/ibm-operator-catalog:latest`.

You can disable or re-enable automatic patch upgrade after installation if you change your mind. For more information, see Configuring automatic patch upgrades.

3. Go to **Administration** > **Cluster Settings**. Under **Configuration** > **OperatorHub** > **Sources**, verify that the *ibm-operator-catalog* CatalogSource object is present.

7. Verify cluster readiness

Run the following command to switch to your IBM Cloud Pak for AIOps project.

```
oc project <project>
```

Where `<project>` is the project that your IBM Cloud Pak for AIOps installation is deployed in.

Run the prerequisite checker script to verify whether your Red Hat OpenShift cluster is correctly set up for a IBM Cloud Pak for AIOps installation. For more information about the script, and how to download and run it, see github.com/IBM.

Example output:

```
# ./prereq.sh
Using project "cp4aiops" on server "https://myserver.mycluster.mydomain:6443".

Starting IBM Cloud Pak for AIOps prerequisite checker v4.2...

[INFO] ================================Openshift Container Platform Version
Check================================
[INFO] Checking OCP Version. Compatible Versions of OCP is v4.12.x
[INFO] OCP version 4.12.22 is compaitble
[INFO] ================================Openshift Container Platform Version
Check================================

[INFO] ================================Entitlement Pull
Secret================================
[INFO] Checking whether the Entitlement secret or Global pull secret is configured correctly.
[INFO] Checking if the job 'cp4aiops-entitlement-key-test-job' already exists.
[INFO] The job with name 'cp4aiops-entitlement-key-test-job' was not found, so moving ahead
and creating it.
[INFO] Creating the job 'cp4aiops-entitlement-key-test-job'
job.batch/cp4aiops-entitlement-key-test-job created
[INFO] Verifying if the job 'cp4aiops-entitlement-key-test-job' completed successfully..
[INFO] SUCCESS! Entitlement secret is configured correctly.
job.batch "cp4aiops-entitlement-key-test-job" deleted
[INFO] ================================Entitlement Pull
Secret================================

[INFO] ================================Storage Provider================================
[INFO] Checking storage providers
[INFO] No IBM Storage Fusion Found... Skipping configuration check.
```

```
[INFO] No Portworx StorageClusters found with "Running" or "Online" status. Skipping
configuration check for Portworx.
[INFO] Openshift Data Foundation found.
[INFO] No IBM Cloud Storage found... Skipping configuration check for IBM Cloud Storage Check.

Checking Openshift Data Foundation Configuration...
Verifying if Red Hat Openshift Data Foundation pods are in "Running" or "Completed" status
[INFO] Pods in openshift-storage project are "Running" or "Completed"
[INFO] ocs-storagecluster-ceph-rbd exists.
[INFO] ocs-storagecluster-cephfs exists.
[INFO] No warnings or failures found when checking for Storage Providers.
[INFO] ================================Storage Provider================================

[INFO] ================================Small or Large Profile Install
Resources================================
[INFO] Checking for cluster resources

[INFO] ================================Resource
Summary=========================================================
[INFO]                                        Nodes    |      vCPU       |     Memory(GB)
[INFO] Small profile(available/required)   [  14 / 3 ]   [  123 / 62 ]       [  159 / 140 ]
[INFO] Large profile(available/required)   [  14 / 10 ]  [  123 / 162 ]      [  159 / 372 ]
[INFO] ================================Resource
Summary=========================================================
[INFO] Cluster currently has resources available to create a small profile of Cloud Pak for
AIOps

[INFO] ================================Small or Large Profile Install
Resources================================


[INFO] ================================Prerequisite Checker Tool
Summary================================
      [  PASS  ] Openshift Container Platform Version Check
      [  PASS  ] Entitlement Pull Secret
      [  PASS  ] Storage Provider
      [  PASS  ] Small or Large Profile Install Resources
[INFO] ================================Prerequisite Checker Tool
Summary================================
```

8. Install the operator

For more information about installing operators on a cluster, see the Red Hat OpenShift documentation Adding Operators.

For more information about the operators which are installed with IBM Cloud Pak for AIOps, see Operator Details.

1. Log in to your Red Hat OpenShift cluster's console.
2. Click **Operators** > **OperatorHub**. The **OperatorHub** page is displayed.
3. In the **All Items** field, enter `IBM Cloud Pak for AIOps`. The **IBM Cloud Pak for AIOps** operator is displayed.
4. Click the **IBM Cloud Pak for AIOps** tile. The **IBM Cloud Pak for AIOps** window is displayed.
5. Click **Install**. The **Install Operator** page is displayed.
6. Enter the following values:
   - Set **Update channel** to *v4.2*.
   - **Installation mode** - For more information about installation modes, see Operator installation mode.
   - **Installed Namespace** - If you are using the `OwnNamespace` installation mode (a specific namespace), then set this field to be the project (namespace) in which to install the operator, such as `cp4aiops`. If you are using the `AllNamespaces` installation mode, then set this field to `openshift-operators`.
   - Set **Update approval** to *Automatic*.

   Warning: Update approval must not be changed to *Manual*. Manual approval, which requires the manual review and approval of the generated InstallPlans, is not supported. Incorrect timing or ordering of manual approvals of

InstallPlans can result in a failed installation.

7. Click **Install** and wait for the IBM Cloud Pak for AIOps operator to install.
8. Verify that the IBM Cloud Pak for AIOps operator is successfully installed.

   Navigate to **Operators** > **Installed Operators**, and select your project from the **Projects** dropdown. **IBM Cloud Pak for AIOps** and its dependant operators in the project are listed with a status of *Succeeded*.

9. Create the custom resource

Create an instance of the IBM Cloud Pak for AIOps custom resource. A maximum of one IBM Cloud Pak for AIOps custom resource is allowed per cluster.

1. From your Red Hat OpenShift console, click **Operators** > **Installed Operators**.
2. From the **Project** dropdown menu, select the project that you want to create the IBM Cloud Pak for AIOps instance in. Use the project that you created earlier in Create a custom project (namespace).

   Note: You cannot use the `default`, `kube-system`, `kube-public`, `openshift-node`, `openshift-infra`, or `openshift` projects. This is because IBM Cloud Pak for AIOps uses Security Context Constraints (SCC), and SCCs cannot be assigned to pods created in one of the default Red Hat OpenShift projects.
3. Select **IBM Cloud Pak for AIOps** operator, then click the **IBM Cloud Pak for AIOps** tab.
4. Click **Create Installation**. The default **Form View** is displayed.

   Warning: The pakModules `aiopsFoundation`, `applicationManager`, and `aiManager` must be enabled. Do not change these values to false.

   Enter the following values:

   - **Name**: Specify the name that you want your IBM Cloud Pak for AIOps instance to be called.
   - **License**: Expand the **License** section and read the agreement. Toggle the **License Acceptance** switch to **True** to accept the license.
   - **File Storage Class** and **Large Block Storage Class** are the storage classes that you want to use, as detailed in the following table. For more information about storage, see Storage.
   - **Enable Secure Tunnel**: Set to true if you want to install Secure Tunnel. For more information about Secure Tunnel, see Secure Tunnel.
   - **Image Pull Secret**: Select the `ibm-entitlement-key` secret that you created in the step Create the entitlement key secret.
   - **Resource Overrides ConfigMap** Do not edit this field unless you have been supplied with a custom `ConfigMap` by an IBM Sales representative.
   - **Size**: Select the size that you require for your IBM Cloud Pak for AIOps installation.
   - **Topology resource group terminology**: Specify `application` or `service` as the terminology to be used for collections of topology resource groups. The default is `application`.

     Note: To confirm that you have the storage classes for your chosen storage provider as shown in the table, run `oc get sc`.

Table. Storage provider classes

Storage provider	File Storage Class	Large Block Storage Class
IBM Cloud®	ibmc-file-gold-gid	ibmc-block-gold
Red Hat® OpenShift® Data Foundation	ocs-storagecluster-cephfs	ocs-storagecluster-ceph-rbd
IBM Storage Fusion	ibm-spectrum-scale-sc	ibm-spectrum-scale-sc
IBM Spectrum Scale Container Native	ibm-spectrum-scale-sc	ibm-spectrum-scale-sc

Storage provider	File Storage Class	Large Block Storage Class
Portworx	portworx-fs	portworx-aiops

5. Click **Create** to create a custom resource that is an instance of IBM Cloud Pak for AIOps.
6. After a few minutes, use the following steps to check the status of your installation. Click **Operators** > **Installed Operators**.
7. From the **Project** list, select the project (namespace) that IBM Cloud Pak for AIOps is deployed in.
8. Select **IBM Cloud Pak for AIOps** and then click the **IBM Cloud Pak for AIOps** tab.
9. Under **Installations**, look for the entry with the name that you specified for your IBM Cloud Pak for AIOps instance, and verify that it has a **Status** of *Phase: Updating*. It takes around 60-90 minutes for the installation to complete (subject to the speed with which images can be pulled). When installation is complete and successful, the **Status** changes to *Phase: Running*.

   (Optional): If you want to see more detail about the status of the installation's components, select the entry with the name that you specified for your IBM Cloud Pak for AIOps instance, and then switch to the **YAML** view. Scroll down to the **Status** section near the end of the YAML. A component's installation is complete and successful when the component has a value of **`Ready`**.

   Example YAML:

```
status:
  size: small
  customProfileConfigmap: aiops-custom-size-profile
  customProfileValidationStatus: >-
    Custom profile configmap not found, continue installation process without
customization
  storageclasslargeblock: rook-ceph-rbd
  componentstatus:
    issueresolutioncore: Ready
    kafka: Ready
    aiopsanalyticsorchestrator: Ready
    aiopsedge: Ready
    tunnel: Ready
    lifecycleservice: Ready
    zenservice: Ready
    flinkcluster: Ready
    cluster: Ready
    elasticsearch: Ready
    kong: Ready
    aiopsui: Ready
    redissentinel: Ready
    <...>
```

(Optional) You can also download and run a status checker script to see information about the status of your deployment. For more information about how to download and run the script, see github.com/IBM.

If the installation fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade and Known Issues to help you identify any installation problems

10. Create an EgressFirewall

There is no egress firewall policy defined when you install IBM Cloud Pak for AIOps, so outgoing traffic from workload pods to the internal and external network is unrestricted.

To create a more secure environment, use the following steps.

1. Create an **`EgressFirewall`** on your Red Hat OpenShift cluster to limit egress from the IBM Cloud Pak for AIOps project (namespace).

   For more information about creating an **`EgressFirewall`**, see Configuring an egress firewall for a project.

   Note: You can only have one **`EgressFirewall`** per project/namespace.

2. Configure exceptions to the **`EgressFirewall`**.

   Edit your **`EgressFirewall`** to add exceptions for the following IBM Cloud Pak for AIOps components that have egress dependencies, otherwise these IBM Cloud Pak for AIOps components fail when they attempt egress.

   1. Allow egress to any external services, such as the following connections:
      - Kubernetes
      - GitHub
      - Microsoft® Teams
      - ServiceNow
      - Slack
      - VMware® vCenter
   2. Configure your **`EgressFirewall`** to allow traffic for your GitHub, Kubernetes, ServiceNow, and VMware vCenter integrations.

      Edit your **`EgressFirewall`** to allow or deny egress, as in the following example:

      ```
      kind: EgressFirewall
      metadata:
        name: default
      spec:
        egress:
        - type: Allow
          to:
            cidrSelector: <1.2.3.0/24>
        - type: Allow
          to:
            dnsName: <www.github.com>
        - type: Allow
          to:
            dnsName: <www.developer.kubernetes.com>
        - type: Allow
          to:
            dnsName: <www.developer.servicenow.com>
        - type: Allow
          to:
            dnsName: <www.developer.vcenter.com>
        - type: Deny
          to:
            cidrSelector: <0.0.0.0/0>
      ```

      Where the values you enter for **`dnsName`** and **`cidrSelector`** are the DNS names and addresses of your GitHub, Kubernetes, ServiceNow, or VMware vCenter sources.

11. Access the IBM Cloud Pak for AIOps UI

After you successfully install IBM Cloud Pak for AIOps, you can use the IBM Cloud Pak Administration panel to manage the underlying deployment, or use the IBM Cloud Pak Automation console to use IBM Cloud Pak for AIOps. You can log in to the console by using **OpenShift authentication** or **IBM provided credentials**.

IBM Cloud Pak Administration panel

You can use the **Launch Admin Hub** link to access the IBM Cloud Pak Administration panel:

1. Log in to the Red Hat OpenShift Container Platform web console as an administrator.
2. Click **Operators** > **Installed Operators**.
3. Click **IBM Cloud Pak for AIOps**.
4. On the *Operator Details* page, click the **IBM Cloud Pak for AIOps** tab, and then click the IBM Cloud Pak for AIOps installation name.
5. In the Details tab, right-click on the URL underneath **Launch Admin Hub**, and select **Open Link in New Tab**.
6. On the IBM Cloud Pak Administration panel login page, select one of the following login options:

- **OpenShift authentication**: The `kubeadmin` user is automatically used to log in to the Administration panel. The `kubeadmin` user has the same privileges as the Administration panel admin user.
- **IBM provided credentials (admin only)**: The default username to access the console is `admin`. To obtain the username and password, see Obtain IBM provided credentials (admin only).

Automation console

You can use the **Launch Cloud Pak in IBM Automation** link to access the Automation console:

1. Log in to the Red Hat OpenShift Container Platform web console as an administrator.
2. Click **Operators** > **Installed Operators**.
3. Click **IBM Cloud Pak for AIOps**.
4. On the *Operator Details* page, click the **IBM Cloud Pak for AIOps** tab, and then click the IBM Cloud Pak for AIOps installation name.
5. In the **Details** tab, right-click on the URL underneath **Launch Cloud Pak in IBM Automation**, and select **Open Link in New Tab**.
6. In the Automation console login page, select one of the following login options:
   - **OpenShift authentication**: The `kubeadmin` user is automatically used to log in to the Automation console. The `kubeadmin` user has the same privileges as the Automation console admin user.
   - **IBM provided credentials (admin only)**: The default username to access the console is `admin`. To obtain the username and password, see Obtain IBM provided credentials (admin only).
   - **Enterprise LDAP**: LDAP users can log in to the Automation console after IBM Cloud Pak for AIOps is configured with a single or multiple LDAP servers for the authentication and authorization. For more information, see Configuring LDAP servers and user authentication

Obtain IBM provided credentials (admin only)

1. To find the default username, run the following command:

   ```
   oc -n ibm-common-services get secret platform-auth-idp-credentials -o
   jsonpath='{.data.admin_username}' | base64 -d && echo
   ```

2. To get the password for the *admin* username, run the following command:

   ```
   oc -n ibm-common-services get secret platform-auth-idp-credentials -o
   jsonpath='{.data.admin_password}' | base64 -d
   ```

   The following is a sample output:

   ```
   EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA
   ```

   Based on the sample output, your password would be `EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA`.

   Important: You can change this default password at any time. For more information, see Changing the cluster administrator password.

What to do next

- Define integrations and applications with Defining.
- If you have an existing on-premises IBM Tivoli Netcool/OMNIbus deployment, then you can connect IBM Cloud Pak for AIOps to it with the Netcool connector. For more information, see Creating IBM Tivoli Netcool/OMNIbus connections.
- If you have an existing on-premises IBM Tivoli Netcool/Impact deployment, then you can connect IBM Cloud Pak for AIOps to it with the IBM Tivoli Netcool/Impact connector. For more information, see Creating IBM Tivoli Netcool/Impact connections
- Familiarize yourself with backup and restore procedures. It is recommended that you take regular backups of your IBM Cloud Pak for AIOps deployment. For more information, see Backup and restore.

Online production installation of IBM Cloud Pak for AIOps (CLI method)

If your cluster is connected to the internet, you can complete a production installation of IBM Cloud Pak® for AIOps with the Red Hat® OpenShift® Container Platform command line interface (CLI).

Before you begin

- Review the Planning section.
- Online installations of IBM Cloud Pak for AIOps can be run entirely as a non-root user, and do not require that user to have sudo access.
- Ensure that you are logged in to your Red Hat OpenShift cluster with `oc login` for any steps that use the Red Hat OpenShift command-line interface (CLI).
- If you require details about the permissions that the IBM Cloud Pak for AIOps operators need, see Permissions (IBM Cloud Pak for AIOps).
- Red Hat OpenShift requires a user with `cluster-admin` privileges for the following operations:
  - Create a custom project
  - Create the catalog source
  - Create an OperatorGroup

Installation procedure

Follow these steps to install IBM Cloud Pak for AIOps.

1. Install and configure Red Hat OpenShift
2. Configure storage
3. Retrieve your entitlement key
4. Create environment variables
5. Create a custom project (namespace)
6. Create the entitlement key secret
7. Configure usage data collection
8. Create the catalog source
9. Verify cluster readiness
10. Install the operator
11. Create the custom resource
12. Create an EgressFirewall to restrict outgoing traffic
13. Access the Automation console

1. Install and configure Red Hat OpenShift

IBM Cloud Pak for AIOps requires Red Hat OpenShift to be installed and running. You must have administrative access to your Red Hat OpenShift cluster.

For more information about the supported versions of Red Hat OpenShift, see Supported Red Hat OpenShift Container Platform versions. The hardware architecture that you install IBM Cloud Pak for AIOps on must be AMD64.

1. Install Red Hat OpenShift by using the instructions in the Red Hat OpenShift documentation.

2. Install the Red Hat OpenShift command line interface (`oc`) on your cluster's boot node and run `oc login`. For more information, see the instructions in Getting started with the Red Hat OpenShift CLI.

3. Ensure that the clocks on your Red Hat OpenShift cluster are synchronized. Each Red Hat OpenShift node in the cluster must have access to an NTP server. Red Hat OpenShift nodes use NTP to synchronize their clocks. IBM Cloud Pak for AIOps runs on Red Hat OpenShift and also has this requirement. Discrepancies between the clocks on the Red Hat OpenShift nodes can cause IBM Cloud Pak for AIOps to experience operational issues. See the Red Hat OpenShift documentation for information about how to use a `MachineConfig` custom resource to configure chrony to connect to your NTP servers.

4. Optionally configure a custom certificate for IBM Cloud Pak for AIOps to use. You can use either of the following methods:

    - Configure a custom certificate for the Red Hat OpenShift cluster. Follow the instructions in the Red Hat OpenShift documentation Replacing the default ingress certificate. Then, deploy the signing CA certificate into the cluster by following the instructions in the Red Hat OpenShift documentation Replacing the CA Bundle certificate.
    - If you would like to use a custom certificate for IBM Cloud Pak for AIOps only, then after installation is complete follow the instructions in Using a custom certificate.

2. Configure storage

Configure persistent storage for IBM Cloud Pak for AIOps that satisfies your sizing requirements. For more information about storage, see Storage.

3. Retrieve your entitlement key

Obtain the IBM entitlement key that is assigned to your **IBMid**, which is needed to pull the IBM Cloud Pak for AIOps images from the IBM® Entitled Registry.

1. Log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.

2. In the *Entitlement key* section, select **Copy** to copy the entitlement key to the clipboard. You will need to use this value to set the environment variable $IBM_ENTITLEMENT_KEY in the next step.

4. Create environment variables

Create and then source a shell script named *waiops_var.sh* that defines the environment variables that will be used to provide installation parameters for your deployment. Use the following codeblock as a template, replacing the brackets **< ... >** with values for your environment.

You can use the following table to find the values to set for your storage environment variables.

Table. Storage provider classes

Storage provider	RWX-storage-class-name	RWO-storage-class-name
IBM Cloud®	ibmc-file-gold-gid	ibmc-block-gold
Red Hat® OpenShift® Data Foundation	ocs-storagecluster-cephfs	ocs-storagecluster-ceph-rbd
IBM Storage Fusion	ibm-spectrum-scale-sc	ibm-spectrum-scale-sc
IBM Spectrum Scale Container Native	ibm-spectrum-scale-sc	ibm-spectrum-scale-sc
Portworx	portworx-fs	portworx-aiops

Note: To confirm that you have the storage classes for your chosen storage provider as shown in the table, run **oc get sc**.

```
#============================================================================================
===========
# Cloud Pak for AIOps installation variables
#============================================================================================
===========

export CP4AIOPS_NAME=ibm-cp-aiops
export CP4AIOPS_SIZE=large # Set to `small` if you only require a starter, non-production
deployment.
export SECURE_TUNNEL=false # Set to `true` to install Secure Tunnel, otherwise set to `false`.
export PROJECT_CP4AIOPS=cp4aiops
export ACCEPT_LICENSE=false # Set to `true` to agree to the license terms, otherwise install
will fail.
export CATALOG_SRC_CP4AIOPS=ibm-operator-catalog

# ------------------------------------------------------------------------------------------
-----------
# Storage
```

```
# ------------------------------------------------------------------------------------------------
-----------
export STG_CLASS=<RWX-storage-class-name>
export STG_CLASS_BLOCK=<RWO-storage-class-name>

# ------------------------------------------------------------------------------------------------
-----------
# Your customer details
# ------------------------------------------------------------------------------------------------
-----------
export CUSTOMER_NAME=<your company name>
export CUSTOMER_ICN=<your IBM Customer Number>
export CUSTOMER_ENVIRONMENT=<Set to `trial`, `poc`, or `production`>

# ------------------------------------------------------------------------------------------------
-----------
# IBM Entitled Registry
# ------------------------------------------------------------------------------------------------
-----------
export IBM_ENTITLEMENT_KEY=<IBM-entitlement-key> # Set to the entitlement key retrieved in
previous step.

# ------------------------------------------------------------------------------------------------
-----------
# `OwnNamespace` installation mode:  leave INSTALL_MODE_NAMESPACE as it is.
# `AllNamespaces` installation mode: change to export INSTALL_MODE_NAMESPACE=openshift-
operators
# ------------------------------------------------------------------------------------------------
-----------
export INSTALL_MODE_NAMESPACE=${PROJECT_CP4AIOPS}

# ------------------------------------------------------------------------------------------------
-----------
# Topology resource group terminology
# Specify `application` or `service` as the terminology to be used for collections of topology
resource
# groups. The default is `application`.
# ------------------------------------------------------------------------------------------------
-----------
export TOPOLOGY_TERMINOLOGY=application
```

If you need help with deciding on the values to set for these environment variables, see the following topics.

- Sizing
- Secure Tunnel
- Operator installation mode

Note: You can set a different value for $PROJECT_CP4AIOPS and $CP4AIOPS_NAME if you want. However, you must not use the **default**, **kube-system**, **kube-public**, **openshift-node**, **openshift-infra**, or **openshift** projects (namespaces) for $PROJECT_CP4AIOPS. This is because IBM Cloud Pak for AIOps uses Security Context Constraints (SCC), and SCCs cannot be assigned to pods created in one of the default Red Hat OpenShift projects (namespaces).

Run the following command to source your script and set the environment variables:

```
. ./waiops_var.sh
```

5. Create a custom project (namespace)

Run the following command to create a project (namespace) to deploy IBM Cloud Pak for AIOps into.

```
oc create namespace "${PROJECT_CP4AIOPS}"
```

6. Create the entitlement key secret

Run the following command to create an image pull secret called *ibm-entitlement-key* to enable your deployment to pull the IBM Cloud Pak for AIOps images from the IBM Entitled Registry.

```
oc create secret docker-registry ibm-entitlement-key --docker-username=cp --docker-
password="${IBM_ENTITLEMENT_KEY}" --docker-server=cp.icr.io --namespace=${PROJECT_CP4AIOPS}
```

7. Configure usage data collection

To help the development of IBM Cloud Pak for AIOps, daily aggregated usage data is collected to analyse how IBM Cloud Pak for AIOps is used. The usage data is collected by the **cp4waiops-metricsprocessor** pod, and is sent to and stored in IBM controlled GDPR-compliant systems. The collection of usage data is enabled by default, but can be disabled. For transparency, the **cp4waiops-metricsprocessor** pod's logs contain all the information that is collected. The usage data that is collected is all numeric, and does not include email addresses, passwords, or specific details. Only the following data is collected:

- Current number of applications
- Current number of alerts (all severities aggregated)
- Current number of incidents (all priorities aggregated)
- Current number of policies (includes predefined and user created)
- Current number of runbooks run since installation
- Current number of connectors of each type (For example ServiceNow, Instana, Falcon Logscale)

Configuring the collection of usage data

If you do not want to disable the collection of usage data, run the following command to configure the usage data with your customer details.

```
oc create secret generic aiops-metrics-processor -n ${PROJECT_CP4AIOPS} --from-
literal=customerName=${CUSTOMER_NAME} --from-literal=customerICN=${CUSTOMER_ICN} --from-
literal=environment=${CUSTOMER_ENVIRONMENT}
```

If you have a firewall enabled, ensure that outbound traffic to https://api.segment.io is allowed.

Important: Usage data without your customer details is still collected even if you do not create this secret. If you do not want any usage data collected, then you must run the command that is given in *Disabling the collection of usage data*.

Disabling the collection of usage data

If you want to disable the collection of usage data, run the following command.

```
oc create secret generic aiops-metrics-processor -n ${PROJECT_CP4AIOPS} --from-
literal=customerName=${CUSTOMER_NAME} --from-literal=customerICN=${CUSTOMER_ICN} --from-
literal=environment=${CUSTOMER_ENVIRONMENT} --from-literal=enableCollection=false
```

Note: You can update your usage data collection preferences after installation. For more information, see Updating usage data collection preferences.

8. Create the catalog source

Add the IBM Cloud Pak for AIOps catalog source to your Red Hat OpenShift cluster.

After installation, the **ibm-operator-catalog** CatalogSource object determines whether the upgrade of your IBM Cloud Pak for AIOps deployment is initiated automatically when a new patch becomes available. The **ibm-operator-catalog** CatalogSource object can be configured to automatically poll for and retrieve a newer catalog by enabling the polling attribute, **spec.updateStrategy.registryPoll**. If a newer catalog for a patch is found and retrieved, then an automatic upgrade of your IBM Cloud Pak for AIOps deployment is initiated. For more information, see Controlling upgrade.

Note: **ibm-operator-catalog** also contains the catalogs for other IBM Cloud Paks®. If multiple IBM Cloud Paks are installed on your cluster, then the polling attribute is configured for all of them.

1. Add the **IBM Operators** CatalogSource.

   If you want to enable the automatic initiation of patch upgrades, run the following command to create the **ibm-operator-catalog** CatalogSource object with polling enabled.

   ```
   cat << EOF | oc apply -f -
   apiVersion: operators.coreos.com/v1alpha1
   kind: CatalogSource
   metadata:
   ```

```
  name: ${CATALOG_SRC_CP4AIOPS}
  namespace: openshift-marketplace
spec:
  displayName: ${CATALOG_SRC_CP4AIOPS}
  publisher: IBM Content
  sourceType: grpc
  image: icr.io/cpopen/${CATALOG_SRC_CP4AIOPS}:latest
  updateStrategy:
    registryPoll:
      interval: 45m
EOF
```

If you want to disable the automatic initiation of patch upgrades, then use the following steps.

1. Run the following command to create the **ibm-operator-catalog** CatalogSource object without polling enabled.

   ```
   cat << EOF | oc apply -f -
   apiVersion: operators.coreos.com/v1alpha1
   kind: CatalogSource
   metadata:
     name: ${CATALOG_SRC_CP4AIOPS}
     namespace: openshift-marketplace
   spec:
     displayName: ${CATALOG_SRC_CP4AIOPS}
     publisher: IBM Content
     sourceType: grpc
     image: icr.io/cpopen/${CATALOG_SRC_CP4AIOPS}:latest
   EOF
   ```

2. Update the **ibm-operator-catalog** CatalogSource to fix it to always use the current image digest, instead of **icr.io/cpopen/ibm-operator-catalog:latest**. This ensures that the **ibm-operator-catalog** CatalogSource pods do not pull the latest image if a node reload or other issue causes them to restart. Run the following commands:

   ```
   IMGDIGEST=`oc get pods -n openshift-marketplace -l=olm.catalogSource=ibm-operator-
   catalog --no-headers -o=jsonpath="{.items[0].status.containerStatuses[0].imageID}" -
   n openshift-marketplace`
   CATALOGIMG=`oc get catalogsource ibm-operator-catalog -n openshift-marketplace -o
   jsonpath='{.spec.image}'`
   if [ "${CATALOGIMG}" = "icr.io/cpopen/ibm-operator-catalog:latest" ]; then
     oc patch catalogsource ibm-operator-catalog -n openshift-marketplace --type=json -
   p "[{ "op": "replace", "path": "/spec/image", "value": "\"$IMGDIGEST\"" }]"
   fi
   ```

You can disable or re-enable automatic patch upgrade after installation if you change your mind. For more information, see Configuring automatic patch upgrades.

2. Verify that the *ibm-operator-catalog* CatalogSource object is present, and is returned by the following command.

   ```
   oc get CatalogSources ibm-operator-catalog -n openshift-marketplace
   ```

   Example output:

   ```
   oc get CatalogSources ibm-operator-catalog -n openshift-marketplace
   NAME                    DISPLAY                 TYPE   PUBLISHER   AGE
   ibm-operator-catalog    IBM Operator Catalog    grpc   IBM         4h13m
   ```

9. Verify cluster readiness

Run the following command to switch to your IBM Cloud Pak for AIOps project.

```
oc project ${PROJECT_CP4AIOPS}
```

Run the prerequisite checker script to verify whether your Red Hat OpenShift cluster is correctly set up for a IBM Cloud Pak for AIOps installation. For more information about the script, and how to download and run it, see github.com/IBM.

Example output:

```
# ./prereq.sh
Using project "cp4aiops" on server "https://myserver.mycluster.mydomain:6443".

Starting IBM Cloud Pak for AIOps prerequisite checker v4.2...

[INFO] ================================Openshift Container Platform Version
Check================================
[INFO] Checking OCP Version. Compatible Versions of OCP is v4.12.x
[INFO] OCP version 4.12.22 is compaitble
[INFO] ================================Openshift Container Platform Version
Check================================

[INFO] ================================Entitlement Pull
Secret================================
[INFO] Checking whether the Entitlement secret or Global pull secret is configured correctly.
[INFO] Checking if the job 'cp4aiops-entitlement-key-test-job' already exists.
[INFO] The job with name 'cp4aiops-entitlement-key-test-job' was not found, so moving ahead
and creating it.
[INFO] Creating the job 'cp4aiops-entitlement-key-test-job'
job.batch/cp4aiops-entitlement-key-test-job created
[INFO] Verifying if the job 'cp4aiops-entitlement-key-test-job' completed successfully..
[INFO] SUCCESS! Entitlement secret is configured correctly.
job.batch "cp4aiops-entitlement-key-test-job" deleted
[INFO] ================================Entitlement Pull
Secret================================

[INFO] ================================Storage Provider================================
[INFO] Checking storage providers
[INFO] No IBM Storage Fusion Found... Skipping configuration check.

[INFO] No Portworx StorageClusters found with "Running" or "Online" status. Skipping
configuration check for Portworx.
[INFO] Openshift Data Foundation found.
[INFO] No IBM Cloud Storage found... Skipping configuration check for IBM Cloud Storage Check.

Checking Openshift Data Foundation Configuration...
Verifying if Red Hat Openshift Data Foundation pods are in "Running" or "Completed" status
[INFO] Pods in openshift-storage project are "Running" or "Completed"
[INFO] ocs-storagecluster-ceph-rbd exists.
[INFO] ocs-storagecluster-cephfs exists.
[INFO] No warnings or failures found when checking for Storage Providers.
[INFO] ================================Storage Provider================================

[INFO] ================================Small or Large Profile Install
Resources================================
[INFO] Checking for cluster resources

[INFO] ================================Resource
Summary=========================================================
[INFO]                                        Nodes    |      vCPU      |     Memory(GB)
[INFO] Small profile(available/required)  [  14 / 3 ]   [  123 / 62 ]       [  159 / 140 ]
[INFO] Large profile(available/required)  [  14 / 10 ]  [  123 / 162 ]      [  159 / 372 ]
[INFO] ================================Resource
Summary=========================================================
[INFO] Cluster currently has resources available to create a small profile of Cloud Pak for
AIOps

[INFO] ================================Small or Large Profile Install
Resources================================


[INFO] ================================Prerequisite Checker Tool
Summary================================
      [  PASS  ] Openshift Container Platform Version Check
      [  PASS  ] Entitlement Pull Secret
      [  PASS  ] Storage Provider
      [  PASS  ] Small or Large Profile Install Resources
[INFO] ================================Prerequisite Checker Tool
Summary================================
```

10. Install the operator

For more information about installing operators on a cluster, see the Red Hat OpenShift documentation Adding Operators.

For more information about the operators which are installed with IBM Cloud Pak for AIOps, see Operator Details.

1. Create an OperatorGroup.

   Important: Skip this step if you are installing using the 'All Namespaces' installation mode. Check that you set **INSTALL_MODE_NAMESPACE** correctly in step 4, and proceed to the next step, Install the IBM Cloud Pak for AIOps operator.

   If you are installing using the 'OwnNamespace' installation mode, then you must create an operator group in your custom project (namespace), or the IBM Cloud Pak for AIOps operator will not install. There might be an operator group for managing a namespace for given APIs. If there is an operator group for the namespace, do not create a second one.

   Create the operator group by running the following command:

   ```
   cat << EOF | oc apply -f -
   apiVersion: operators.coreos.com/v1
   kind: OperatorGroup
   metadata:
     name: cp4aiops-operator-group
     namespace: ${PROJECT_CP4AIOPS}
   spec:
     targetNamespaces:
       - "${PROJECT_CP4AIOPS}"
   EOF
   ```

2. Install the IBM Cloud Pak for AIOps operator.

   Run the following command.

   ```
   cat << EOF | oc apply -f -
   apiVersion: operators.coreos.com/v1alpha1
   kind: Subscription
   metadata:
     name: ibm-aiops-orchestrator
     namespace: $INSTALL_MODE_NAMESPACE
   spec:
     channel: v4.2
     installPlanApproval: Automatic
     name: ibm-aiops-orchestrator
     source: ${CATALOG_SRC_CP4AIOPS}
     sourceNamespace: openshift-marketplace
   EOF
   ```

   Warning: installPlanApproval must not be changed to *Manual*. Manual approval, which requires the manual review and approval of the generated InstallPlans, is not supported. Incorrect timing or ordering of manual approvals of InstallPlans can result in a failed installation.

3. After a few minutes, the IBM Cloud Pak for AIOps operator is installed. Verify that the all of the components have a state of **Succeeded** by running the following command:

   ```
   oc get csv -n ${INSTALL_MODE_NAMESPACE} | egrep "ibm-aiops-orchestrator|ibm-common-service"
   ```

   Example output:

   ```
   $ oc get csv -n ${INSTALL_MODE_NAMESPACE} | egrep "ibm-aiops-orchestrator|ibm-common-service"
   ibm-aiops-orchestrator.v4.2.0          IBM Cloud Pak for AIOps                4.2.0
   Succeeded
   ibm-common-service-operator.v3.23.7    IBM Cloud Pak foundational services    3.23.7
   Succeeded
   ```

11. Create the custom resource

Create an instance of the IBM Cloud Pak for AIOps custom resource. A maximum of one IBM Cloud Pak for AIOps custom resource is allowed per cluster.

1. Use the following YAML to create an instance of the IBM Cloud Pak for AIOps custom resource.

```
cat << EOF | oc apply -f -
apiVersion: orchestrator.aiops.ibm.com/v1alpha1
kind: Installation
metadata:
  name: ${CP4AIOPS_NAME}
  namespace: ${PROJECT_CP4AIOPS}
spec:
  size: ${CP4AIOPS_SIZE}
  storageClass: ${STG_CLASS}
  storageClassLargeBlock: ${STG_CLASS_BLOCK}
  imagePullSecret: ibm-entitlement-key
  topologyModel: ${TOPOLOGY_TERMINOLOGY}
  license:
    accept: ${ACCEPT_LICENSE}
  pakModules:
  - name: aiopsFoundation
    enabled: true
  - name: applicationManager
    enabled: true
  - name: aiManager
    enabled: true
  - name: connection
    enabled: ${SECURE_TUNNEL}
EOF
```

Warning: The pakModules **aiopsFoundation**, **applicationManager**, and **aiManager** must be enabled as in the following YAML. Do not change these values to false.

2. Verify your installation.

   Run the following command to check that the **PHASE** of your installation is **Updating**.

```
oc get installations.orchestrator.aiops.ibm.com -n ${PROJECT_CP4AIOPS}
```

   Example output:

```
NAME                    PHASE      LICENSE    STORAGECLASS    STORAGECLASSLARGEBLOCK    AGE
ibm-cp-aiops            Updating   Accepted   rook-cephfs     rook-ceph-block           3m
```

   It takes around 60-90 minutes for the installation to complete (subject to the speed with which images can be pulled). When installation is complete and successful, the **PHASE** of your installation changes to **Running**. If your installation phase does not change to **Running**, then use the following command to find out which components are not ready:

```
oc get installation.orchestrator.aiops.ibm.com -o yaml -n ${PROJECT_CP4AIOPS} | grep 'Not
Ready'
```

   Example output:

```
lifecycleservice: Not Ready
zenservice: Not Ready
```

   To see details about why a component is **Not Ready** run the following command, where **<component>** is the component that is not ready, for example **zenservice**.

```
oc get <component> -o yaml -n ${PROJECT_CP4AIOPS}
```

   (Optional) You can also download and run a status checker script to see information about the status of your deployment. For more information about how to download and run the script, see github.com/IBM.

If the installation fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade and Known Issues to help you identify any installation problems.

12. Create an EgressFirewall

There is no egress firewall policy defined when you install IBM Cloud Pak for AIOps, so outgoing traffic from workload pods to the internal and external network is unrestricted.

To create a more secure environment, use the following steps.

1. Create an **`EgressFirewall`** on your Red Hat OpenShift cluster to limit egress from the IBM Cloud Pak for AIOps project (namespace).

   For more information about creating an **`EgressFirewall`**, see Configuring an egress firewall for a project.

   Note: You can have only one **`EgressFirewall`** per project/namespace.

2. Configure exceptions to the **`EgressFirewall`**.

   Edit your **`EgressFirewall`** to add exceptions for the following IBM Cloud Pak for AIOps components that have egress dependencies, otherwise these components fail when they attempt egress.

   1. Allow egress to any external services, such as the following connections:
      - Kubernetes
      - GitHub
      - Microsoft® Teams
      - ServiceNow
      - Slack
      - VMware® vCenter
   2. Configure your **`EgressFirewall`** to allow traffic for your GitHub, Kubernetes, ServiceNow, and VMware vCenter integrations.

      Edit your **`EgressFirewall`** to allow or deny egress, as in the following example. Substitute values for **`dnsName`** and **`cidrSelector`** that are the DNS names and addresses of your GitHub, Kubernetes, ServiceNow, or VMware vCenter sources.

```
kind: EgressFirewall
metadata:
  name: default
spec:
  egress:
  - type: Allow
    to:
      cidrSelector: <1.2.3.0/24>
  - type: Allow
    to:
      dnsName: <www.github.com>
  - type: Allow
    to:
      dnsName: <www.developer.kubernetes.com>
  - type: Allow
    to:
      dnsName: <www.developer.servicenow.com>
  - type: Allow
    to:
      dnsName: <www.developer.vcenter.com>
  - type: Deny
    to:
      cidrSelector: <0.0.0.0/0>
```

13. Access the Automation console

After you successfully install IBM Cloud Pak for AIOps, get the URL for accessing the Automation console.

1. Use the following command to get the URL to access the Automation console:

```
oc get route -n ${PROJECT_CP4AIOPS} cpd -o jsonpath='{.spec.host}'
```

The following output is a sample output:

```
cpd-cp4aiops.apps.mycluster.mydomain
```

Based on the sample output, your console URL would be **https://cpd-cp4aiops.apps.mycluster.mydomain**

2. Enter the URL in your browser to open the Automation console. Log in with your username and password.

Find the IBM Cloud Pak for AIOps console username and password

The default username to access the Automation console is **admin**. You can check the default username and their password with the following commands.

Note: This information is for the **IBM provided credentials (admin only)** authentication type.

1. Find the default username.

```
oc -n ibm-common-services get secret platform-auth-idp-credentials -o
jsonpath='{.data.admin_username}' | base64 -d && echo
```

2. Get the password for the **admin** username.

```
oc -n ibm-common-services get secret platform-auth-idp-credentials -o
jsonpath='{.data.admin_password}' | base64 -d
```

The following extract shows a sample output:

```
EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA
```

Based on the sample output, your password would be **EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA**.

Important: You can change this default password at any time. For more information, see Changing the cluster administrator password.

What to do next

- Define integrations and applications with Defining.
- If you have an existing on-premises IBM Tivoli Netcool/OMNIbus deployment, then you can connect IBM Cloud Pak for AIOps to it with the Netcool connector. For more information, see Creating IBM Tivoli Netcool/OMNIbus connections.
- If you have an existing on-premises IBM Tivoli Netcool/Impact deployment, then you can connect IBM Cloud Pak for AIOps to it with the IBM Tivoli Netcool/Impact connector. For more information, see Creating IBM Tivoli Netcool/Impact connections
- Familiarize yourself with backup and restore procedures. It is recommended that you take regular backups of your IBM Cloud Pak for AIOps deployment. For more information, see Backup and restore.

Installing IBM Cloud Pak for AIOps in an air-gapped environment (offline)

If your cluster is not connected to the internet, you can install IBM Cloud Pak® for AIOps in your cluster by using either a bastion host, portable compute device, or a portable storage device.

The IBM Catalog Management Plug-in for IBM Cloud Pak® is used for the deployment of IBM Cloud Paks® in a disconnected environment. It simplifies the process for discovering IBM product images, uses standard tools for registry and cluster access, and extends the Red Hat OpenShift CLI (**oc**) capability to streamline the air-gapped deployment process.

Overview

It is common in production to have a cluster that does not have internet access. In these cases, you can still install IBM Cloud Pak for AIOps and Red Hat® OpenShift® Container Platform in an air-gapped (otherwise known as offline or disconnected) environment. Unlike online installations, air-gapped installations require you to enable the IBM Operator Catalog to mimic a typical online installation by using images in your own registry.

The following diagram provides an overview of air-gapped installation scenarios for portable compute, portable storage, and bastion hosts:

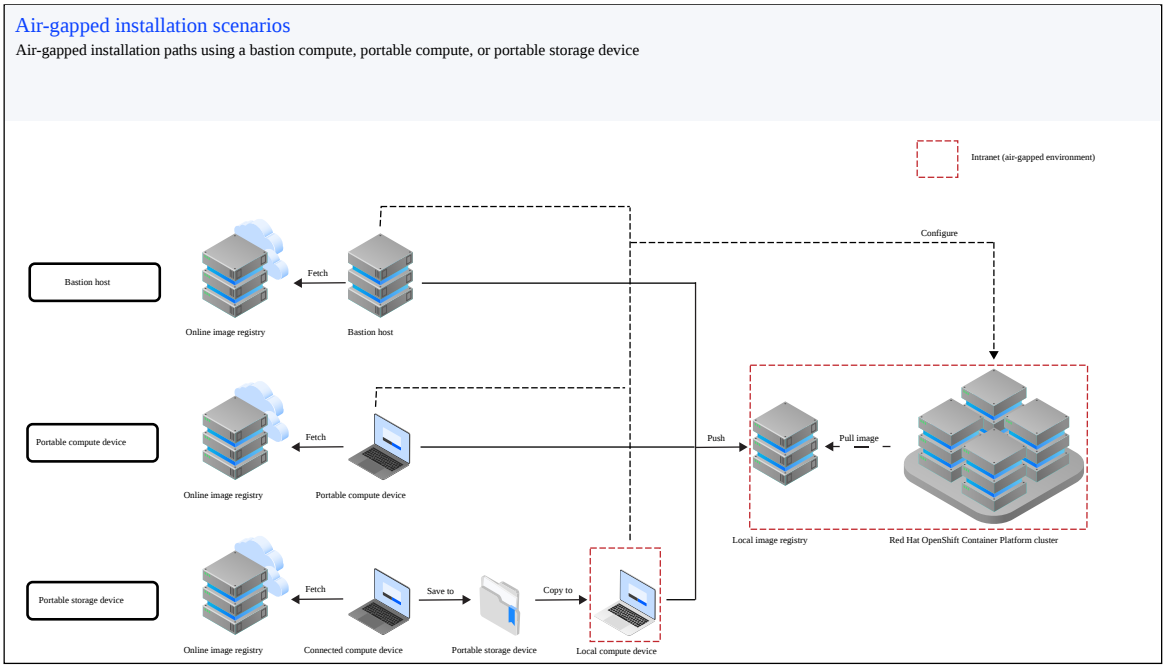


Figure 1. Air-gap scenario

All of these scenarios use Container Application Software for Enterprises (CASE) files. CASE is a specification that defines the metadata and structure for packaging, managing, and unpacking containerized applications.

You can store the product code and images to a portable compute device (like a laptop), portable storage device (like an external hard disk drive), or a bastion host and then transfer them to a local air-gapped network. The images that are stored are large, and you must ensure that each of your airgap devices have enough storage, as follows.

- Bastion host method - the local target registry must have at least 148 GB of storage space.
- Portable device methods - the portable device and the local target registry must each have at least 148 GB of storage space.

Table 1. Air-gapped installation scenarios

Air-gapped installation approach	Example	Description
Bastion host	Bastion host	The bastion host has access to the public internet and the air-gapped environment where a local target registry and a Red Hat OpenShift Container Platform cluster are. The IBM Cloud Pak for AIOps images are replicated from the IBM Entitled Registry through the bastion server directly to the local target registry behind the firewall.
Portable compute device	Laptop	IBM Cloud Pak for AIOps images are downloaded from the IBM Entitled Registry to a file system on a portable compute device. The portable compute device is then disconnected from the internet and brought into the air-gapped environment so that the images can be copied from the file system to the local target registry behind the firewall.
Portable storage device	Portable hard disk drive	IBM Cloud Pak for AIOps images are downloaded from the IBM Entitled Registry through an internet connected device to a file system on a portable storage device. The portable storage device is then disconnected and brought into the air-gapped environment so that the images can be copied from the file system to the local target registry behind the firewall.

Installation methods

Choose one of the following deployment methods to complete an air-gapped installation.

- Installing IBM Cloud Pak for AIOps in an air-gapped environment (offline) using a bastion host
- Installing IBM Cloud Pak for AIOps in an air-gapped environment (offline) using a portable device

Installing IBM Cloud Pak for AIOps in an air-gapped environment (offline) using a bastion host

If your cluster is not connected to the internet, you can deploy a production installation of IBM Cloud Pak® for AIOps on your Red Hat® OpenShift® Container Platform cluster by using a bastion host.

In this scenario, your air-gapped (offline) environment has a target registry, and a Red Hat OpenShift cluster on which IBM Cloud Pak for AIOps is to be installed. The bastion host has access to the internet and to the offline environment. IBM Cloud Pak for AIOps images are mirrored from the internet through the bastion server to the target registry in the offline environment. IBM Cloud Pak for AIOps can then be installed in the offline environment by using the target registry.

Before you begin

- Review the Planning section. Your environment must meet the system requirements.
- If you want to install IBM Cloud Pak for AIOps as a non-root user, you must review the information in Install commands requiring root or sudo.
- Ensure that you are logged in to your Red Hat OpenShift cluster with `oc login` for any steps that use the Red Hat OpenShift command-line interface (CLI).
- If you require details about the permissions that are needed by the IBM Cloud Pak for AIOps operators, see Permissions (IBM Cloud Pak for AIOps).
- Red Hat OpenShift requires a user with `cluster-admin` privileges for the following operations:
  - Create a custom project
  - Configure the cluster
  - Create the catalog source
  - Create an OperatorGroup

Important: The following procedure is based on an Red Hat OpenShift 4.12 environment and includes links for that version. If your environment uses a different supported version of Red Hat OpenShift, ensure that you follow the Red Hat OpenShift documentation for that version.

Installation procedure

1. Set up the mirroring environment
2. Download the CASE
3. Mirror images
4. Configure storage
5. Install IBM Cloud Pak for AIOps

1. Set up the mirroring environment

Prerequisites

Allow access to the following sites and ports:

Table 1. Sites and ports that must be accessible

Site	Description

Site	Description
• `icr.io` • `cp.icr.io` • `dd0.icr.io` • `dd2.icr.io` • `dd4.icr.io` • `dd6.icr.io`	Allow access to these hosts on port 443 to enable access to the **`IBM Cloud Container Registry`**, CASE OCI artifact, and IBM Cloud Pak® foundational services catalog source.
• `dd1-icr.ibm-zh.com` • `dd3-icr.ibm-zh.com` • `dd5-icr.ibm-zh.com` • `dd7-icr.ibm-zh.com`	If you are located in China, also allow access to these hosts on port 443.
`github.com`	Github houses CASE files, IBM Cloud Pak tools and scripts.
`redhat.com`	Red Hat OpenShift registries that are required for Red Hat OpenShift, and for Red Hat OpenShift upgrades. For more information, see Configuring your firewall for OpenShift Container Platform.

1.1 Download documentation and scripts for offline access

Download the following documentation and scripts that you might need to access during your IBM Cloud Pak for AIOps installation, and copy them to your air-gapped environment.

1. IBM Cloud Pak for AIOps 4.2.0 documentation

   Download the Cloud Pak for AIOps 4.2.0 PDF (this documentation) so that you can access it offline.

2. IBM Cloud Pak for AIOps 4.2.0 scripts

   - The prerequisite checker script verifies whether your Red Hat OpenShift cluster is correctly set up for a IBM Cloud Pak for AIOps installation. You will need to run this script in step 5.5 Verify cluster readiness. Download this script from github.com/IBM.
   - An uninstall script can be downloaded from github.com/IBM.
   - (Optional) The status checker script can be used in step 5.7 Create the custom resource to give information about the status of your deployment. The use of this script is optional, as status can be found directly from the **`ibm-aiops-orchestrator`** custom resource. This script can be downloaded from github.com/IBM.

3. Red Hat OpenShift documentation

   The Red Hat OpenShift documentation can be downloaded for offline access from Red Hat. The **`Security and compliance`**, **`Installing`**, **`CLI Tools`**, **`Images`**, and **`Operators`** sections are referenced by this documentation.

1.2 Install and configure Red Hat OpenShift

IBM Cloud Pak for AIOps requires Red Hat OpenShift to be installed and running on your target cluster. You must have administrative access to your Red Hat OpenShift cluster.

For more information about the supported versions of Red Hat OpenShift, see Supported Red Hat OpenShift Container Platform versions. The hardware architecture that you install IBM Cloud Pak for AIOps on must be AMD64.

1. Install Red Hat OpenShift by using the instructions in the Red Hat OpenShift documentation. Information about installing a cluster in a restricted network is given in Mirroring images for a disconnected installation.

2. Install the Red Hat OpenShift command line interface (`oc`) on your cluster's boot node and run `oc login`. For more information, see the instructions in Getting started with the Red Hat OpenShift CLI.

3. Ensure that the clocks on your Red Hat OpenShift cluster are synchronized. Each Red Hat OpenShift node in the cluster must have access to an NTP server. Red Hat OpenShift nodes use NTP to synchronize their clocks. IBM Cloud Pak for AIOps runs on Red Hat OpenShift and also has this requirement. Discrepancies between the clocks on the Red Hat OpenShift nodes can cause IBM Cloud Pak for AIOps to experience operational issues. See the Red Hat OpenShift documentation for information about how to use a `MachineConfig` custom resource to configure chrony to connect to your NTP servers.

4. Optionally configure a custom certificate for IBM Cloud Pak for AIOps to use. You can use either of the following methods:

    - Configure a custom certificate for the Red Hat OpenShift cluster. Follow the instructions in the Red Hat OpenShift documentation Replacing the default ingress certificate. Then, deploy the signing CA certificate into the cluster by following the instructions in the Red Hat OpenShift documentation Replacing the CA Bundle certificate.
    - If you would like to use a custom certificate for IBM Cloud Pak for AIOps only, then after installation is complete follow the instructions in Using a custom certificate.

1.3 Set up a target registry

You must have a local Docker type production-grade registry available to store the IBM Cloud Pak for AIOps images in. The registry must meet the following requirements:

- supports Docker Manifest V2, Schema 2.
- supports multi-architecture images.
- is accessible from the Red Hat OpenShift cluster nodes.
- allows path separators in the image name.
- you have the username and password for a user who can read from and write to the registry.
- must have 148 GB of storage to hold all the software that is to be transferred to the target registry.

If you do not already have a suitable production-grade registry available, then you must install and configure one. For more information, see About the mirror registry in the Red Hat OpenShift documentation.

Important: Do not use the Red Hat OpenShift image registry as your target registry. The Red Hat OpenShift registry does not support multi-architecture images or path separators in the image name.

1.4 Prepare a host

Prepare a bastion host that can connect to the internet and to the air-gapped network with access to the Red Hat OpenShift cluster and the target registry. Your host must be on a Linux® x86_64 or Mac platform with any operating system that the IBM Cloud Pak® CLI and the Red Hat OpenShift CLI support. If you are on a Windows® platform, you must run the actions in a Linux® x86_64 VM or from a Windows Subsystem for Linux (WSL) terminal.

Complete the following steps on your host.

1. Install Docker or Podman. One of these tools is needed for container management.

    To install Podman, see the Podman Installation Instructions.

    To install Docker (for example, on Red Hat® Enterprise Linux®), run the following commands:

    ```
    yum check-update
    yum install docker
    ```

    Note: Docker is not shipped or supported for Red Hat Enterprise Linux (RHEL) 8. The Podman container engine replaced docker as the preferred, maintained, and supported container runtime of choice for Red Hat Enterprise Linux 8 systems. For more information, see Running containers without Docker in the Red Hat documentation.

2. Install the Red Hat OpenShift CLI tool, `oc`.

`oc` is required for Red Hat OpenShift management. For more information, see Getting started with the OpenShift CLI in the Red Hat OpenShift documentation.

1.5 Install the IBM Catalog Management Plug-in for IBM Cloud Pak®

The IBM Catalog Management Plug-in for IBM Cloud Pak (`ibm-pak-plugin`) is used for the deployment of IBM Cloud Paks® in a disconnected environment. It simplifies the process for discovering required IBM product images and uses standard tools for registry and cluster access. The ibm-pak-plugin also extends the Red Hat OpenShift CLI (`oc`) capability to streamline the process of delivering installation images to the IBM Cloud Pak in an air-gapped environment.

1. Download and install the most recent version of the `ibm-pak-plugin` for your host operating system from github.com/IBM.

2. Run the following command to extract the files.

   ```
   tar -xf oc-ibm_pak-linux-amd64.tar.gz
   ```

3. Run the following command to move the file to the `/usr/local/bin` directory.

   ```
   mv oc-ibm_pak-linux-amd64 /usr/local/bin/oc-ibm_pak
   ```

   Note: If you are installing as a non-root user, you must use sudo.

4. Confirm that the `ibm-pak-plugin` is installed by running the following command.

   ```
   oc ibm-pak --help
   ```

   Expected result: The `ibm-pak-plugin` usage is displayed.

2. Download the CASE

Set environment variables on the bastion host, and connect to the internet so that you can download the IBM Cloud Pak for AIOps CASE files.

Note: Save a copy of your environment variable values to a file that you can use as a reference when you are completing your air-gapped installation tasks.

1. Create the following environment variables.

   ```
   export CASE_NAME=ibm-cp-waiops
   export CASE_VERSION=1.8.0
   export CASE_INVENTORY_SETUP=cpwaiopsSetup
   export TARGET_REGISTRY_HOST=<IP_or_FQDN_of_target_registry>
   export TARGET_REGISTRY_PORT=<port_number_of_target_registry>
   export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
   export TARGET_REGISTRY_USER=<username>
   export TARGET_REGISTRY_PASSWORD=<password>
   ```

   The target registry is the registry where the IBM Cloud Pak for AIOps images are mirrored to, and accessed from by the Red Hat OpenShift cluster, as setup in 1.3 Set up a target registry.

   If your bastion host must connect to the internet through a proxy, then also set the following environment variables.

   ```
   export https_proxy=http://proxy-server-hostname:port
   export http_proxy=http://proxy-server-hostname:port
   ```

2. Connect your host to the internet and disconnect it from the local air-gapped network.

3. Download the IBM Cloud Pak for AIOps installer and image inventory to your host.

   ```
   oc ibm-pak get ${CASE_NAME} --version ${CASE_VERSION}
   ```

   The CASE is downloaded to `~/.ibm-pak/data/cases/$CASE_NAME/$CASE_VERSION`. The logs files are available at `~/.ibm-pak/logs/oc-ibm_pak.log`.

   Note: If you do not specify the CASE version, then the latest CASE is downloaded. The root directory that is used by `ibm-pak-plugin` is `~/.ibm-pak`. If required, the root directory can be configured by setting the `IBMPAK_HOME` environment variable.

3. Mirror images

Complete the following steps to mirror the IBM Cloud Pak for AIOps images to the target registry.

3.1. Generate mirror manifests

Run the following command to generate mirror manifests to be used when mirroring the images to the target registry.

```
oc ibm-pak generate mirror-manifests ${CASE_NAME} ${TARGET_REGISTRY} --version ${CASE_VERSION}
```

A new directory named **~/.ibm-pak/mirror** is created when you issue the **oc ibm-pak generate mirror-manifests** command. The files **images-mapping.txt** and **image-content-source-policy.yaml** are generated at **~/.ibm-pak/data/mirror/$CASE_NAME/$CASE_VERSION**.

3.2. Authenticate with the registries

Log in to the registries to generate an authentication file containing the registry credentials, and then create an environment variable that has the location of the authentication file. This file is used later to enable the **oc image mirror** command to pull the images from the IBM Entitled Registry, and push them to the target registry.

1. Get the authentication credentials for the IBM Entitled Registry.
    1. To obtain the entitlement key that is assigned to your IBMid, log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.
    2. In the *Entitlement keys* section, select **Copy key** to copy the entitlement key.
2. Run the following command to create an environment variable that contains your entitlement key.

    ```
    export ENTITLED_REGISTRY_PASSWORD=<key>
    ```

    Where **<key>** is the entitlement key that you copied in the previous step.
3. Store the authentication credentials for the IBM Entitled Registry and the target registry.

    If you are using Podman, run the following commands:

    ```
    podman login cp.icr.io -u cp -p ${ENTITLED_REGISTRY_PASSWORD}
    podman login ${TARGET_REGISTRY} -u ${TARGET_REGISTRY_USER} -p ${TARGET_REGISTRY_PASSWORD}
    export REGISTRY_AUTH_FILE=${XDG_RUNTIME_DIR}/containers/auth.json
    unset ENTITLED_REGISTRY_PASSWORD
    ```

    Note: The authentication file is usually at **${XDG_RUNTIME_DIR}/containers/auth.json**. For more information, see the Options section in the Podman documentation.

    If you are using Docker, run the following commands:

    ```
    docker login cp.icr.io -u cp -p ${ENTITLED_REGISTRY_PASSWORD}
    docker login ${TARGET_REGISTRY} -u ${TARGET_REGISTRY_USER} -p ${TARGET_REGISTRY_PASSWORD}
    export REGISTRY_AUTH_FILE=$HOME/.docker/config.json
    unset ENTITLED_REGISTRY_PASSWORD
    ```

    Note: The authentication file is usually at **$HOME/.docker/config.json**. For more information, see the Docker documentation.

3.3. Mirror the images

Complete these steps on your host that is connected to both the local Docker or Podman registry and the Red Hat OpenShift cluster.

Mirror images to the target registry.

```
nohup oc image mirror \
-f ~/.ibm-pak/data/mirror/${CASE_NAME}/${CASE_VERSION}/images-mapping.txt \
-a ${REGISTRY_AUTH_FILE} \
--filter-by-os '.*' \
--insecure \
```

```
--skip-multiple-scopes \
--max-per-registry=1 > my-mirror-progress.txt 2>&1 &
```

The UNIX command **nohup** is used to ensure that the mirroring process continues even if there is a loss of network connection, and redirection of output to a file provides improved monitoring and error visibility.

Run the following command if you want to see the progress of the mirroring:

```
tail -f my-mirror-progress.txt
```

Note: If an error occurs during mirroring, the mirror command can be rerun.

3.4 Configure the cluster

1. Log in to your Red Hat OpenShift cluster.

   You can identify your specific **oc login** command by clicking the user menu in the upper left of the Red Hat OpenShift console, and then clicking **Copy Login Command**.

   Example:

   ```
   oc login <server> -u <cluster username> -p <cluster pass>
   ```

2. Update the global image pull secret for your Red Hat OpenShift cluster.

   Follow the steps in the Red Hat OpenShift documentation topic Updating the global cluster pull secret.

   These steps enable your cluster to have authentication credentials in place to pull images from your TARGET_REGISTRY as specified in the **image-content-source-policy.yaml**, which you will apply to your cluster in the next step.

3. Create the **ImageContentSourcePolicy**.

   Run the following command to create the ImageContentSourcePolicy.

   ```
   oc apply -f  ~/.ibm-pak/data/mirror/${CASE_NAME}/${CASE_VERSION}/image-content-source-policy.yaml
   ```

4. Verify that the **ImageContentSourcePolicy** resource is created.

   ```
   oc get imageContentSourcePolicy
   ```

5. Verify your cluster node status.

   ```
   oc get MachineConfigPool -w
   ```

   Important: After the **ImageContentsourcePolicy** and global image pull secret are applied, the configuration of your nodes will be updated sequentially. Wait until all the **MachineConfigPools** are updated before you proceed to the next step.

6. (Optional) If you use an insecure registry, you must add the target registry to the cluster's **insecureRegistries** list.

   ```
   oc patch image.config.openshift.io/cluster --type=merge \
   -p '{"spec":{"registrySources":{"insecureRegistries":["'${TARGET_REGISTRY}'"]}}}'
   ```

4. Configure storage

The storage configuration must satisfy your sizing requirements. Two storage classes are needed for installing IBM Cloud Pak for AIOps. For more information, see Storage.

5. Install IBM Cloud Pak for AIOps

Now that the images are mirrored to your air-gapped environment, you can deploy IBM Cloud Pak for AIOps to that environment. To install IBM Cloud Pak for AIOps, complete the following steps.

- 5.1 Create the catalog source
- 5.2 Create environment variables

- 5.3 Create a custom project
- 5.4 Configure usage data collection
- 5.5 Verify cluster readiness
- 5.6 Install the operator
- 5.7 Create the custom resource
- 5.8 Create an EgressFirewall
- 5.9 Access the Automation console

5.1 Create the catalog source

Now that the images are mirrored to your air-gapped environment, you can deploy IBM Cloud Pak for AIOps to that environment

1. Create and configure the catalog source.

```
oc ibm-pak launch \
${CASE_NAME} \
--version ${CASE_VERSION} \
--action install-catalog \
--inventory ${CASE_INVENTORY_SETUP} \
--namespace openshift-marketplace \
--args "--registry ${TARGET_REGISTRY} --recursive \
--inputDir ~/.ibm-pak/data/cases/${CASE_NAME}/${CASE_VERSION}"
```

2. Verify that the CatalogSource is installed.

   Run the following commands.

```
oc get catalogsource -n openshift-marketplace
oc get pods -n openshift-marketplace
```

Example output:

```
oc get catalogsource -n openshift-marketplace
NAME                                          DISPLAY                                          TYPE
PUBLISHER   AGE
aiops-analytics-operator-catalog              IBM AIOps Analytics Product Catalog              grpc
IBM         25m
aiops-lifecycle-operator-catalog              IBM AIOps Lifecycle Product Catalog              grpc
IBM         25m
certified-operators                           Certified Operators                              grpc
Red Hat     24h
cloud-native-postgresql-catalog               Cloud Native Postgresql Catalog                  grpc
IBM         54m
community-operators                           Community Operators                              grpc
Red Hat     24h
ibm-aiops-ir-core-operators                   IBM AIOps Issue Resolution Core Operators        grpc
IBM         54m
ibm-aiopsedge-case                            aiopsedge index                                  grpc
IBM         25m
ibm-asm-catalog                               IBM Agile Service Manager Catalog                grpc
IBM         25m
ibm-cloud-databases-redis-operator-catalog    ibm-cloud-databases-redis-operator-catalog       grpc
IBM         25m
ibm-cp-waiops-catalog                         IBM Cloud Pak for AIOps Catalog                  grpc
IBM         25m
ibm-cp-waiops-elastic-catalog                 IBM CP4AIOps - Elastic                           grpc
IBM         25m
ibm-cp-waiops-flink-catalog                   IBM CP4AIOps - Flink                             grpc
IBM         25m
ibm-kong-management-catalog                   IBM Kong Management Catalog                      grpc
IBM         25m
ibm-secure-tunnel-operator-catalog            IBM Secure Tunnel operator Catalog               grpc
IBM         25m
ibm-watson-ai-manager-catalog                 ibm-watson-ai-manager-catalog                    grpc
IBM         25m
ibm-watson-aiops-ui-catalog                   IBM WAIOps UI Catalog                            grpc
IBM         25m
opencloud-operators                           IBMCS Operators                                  grpc
IBM         25m
redhat-marketplace                            Red Hat Marketplace                              grpc
```

```
Red Hat     24h
redhat-operators                                Red Hat Operators                                grpc
Red Hat     24h
<br>
oc get pods -n openshift-marketplace
NAME                                                              READY   STATUS      RESTARTS
AGE
0935b4423e67bb338b8687eeee2898674099d8fc158ffb50b8bc1fba0a692g2   0/1     Completed   0
24h
1a64439d8a73c4c057540c5e6f4329f15aeaa15b5cc9f254df9ab9dd7bjqzpt   0/1     Completed   0
24h
6766646cfd55d8f1be544bc968431245406bd8d357bcc6c75efb34bb50bc829   0/1     Completed   0
24h
aiops-analytics-operator-catalog-cvwcc                            1/1     Running     0
3m45s
aiops-lifecycle-operator-catalog-2kv2l                            1/1     Running     0
3m46s
certified-operators-vn8ql                                         1/1     Running     0
86m
cloud-native-postgresql-catalog-m6c42                             1/1     Running     0
3m32s
community-operators-h59qg                                         1/1     Running     0
65m
e370676c5b33f2c041ba1d3a7bdc078b247d7d4e6fd61841e5d197770e9n2sf   0/1     Completed   0
24h
f8ea7cf3f9611828a4809f2eaaf2ae944796f25e2ea7c23bb389c7509cpjxf7   0/1     Completed   0
24h
ibm-aiops-ir-core-operators-k69f9                                 1/1     Running     0
3m49s
ibm-aiopsedge-case-wl2nx                                          1/1     Running     0
3m41s
ibm-asm-catalog-qgz4l                                             1/1     Running     0
3m51s
ibm-cloud-databases-redis-operator-catalog-tp7jd                  1/1     Running     0
3m35s
ibm-cp-waiops-catalog-khjmt                                       1/1     Running     0
3m32s
ibm-cp-waiops-elastic-catalog-ptv77                               1/1     Running     0
3m30s
ibm-cp-waiops-flink-catalog-fhs69                                 1/1     Running     0
3m29s
ibm-kong-management-catalog-mr89f                                 1/1     Running     0
3m56s
ibm-secure-tunnel-operator-catalog-76fvf                          1/1     Running     0
3m39s
ibm-watson-ai-manager-catalog-bz66v                               1/1     Running     0
3m53s
ibm-watson-aiops-ui-catalog-9srhm                                 1/1     Running     0
3m55s
marketplace-operator-7dd7c98f68-2t5r9                             1/1     Running     0
25h
opencloud-operators-79j9k                                         1/1     Running     0
3m43s
redhat-marketplace-wc674                                          1/1     Running     0
24h
redhat-operators-rc4b2                                            1/1     Running     0
6h55m
```

5.2 Create environment variables

Create and then source a shell script named *waiops_var.sh* that defines the environment variables that will be used to provide installation parameters for your deployment. Use the following codeblock as a template, replacing the brackets **<** **...** **>** with values for your environment.

You can use the following table to find the values to set for your storage environment variables.

Table 2. Storage provider classes

Storage provider	RWX-storage-class-name	RWO-storage-class-name

Storage provider	RWX-storage-class-name	RWO-storage-class-name
IBM Cloud®	ibmc-file-gold-gid	ibmc-block-gold
Red Hat® OpenShift® Data Foundation	ocs-storagecluster-cephfs	ocs-storagecluster-ceph-rbd
IBM Storage Fusion	ibm-spectrum-scale-sc	ibm-spectrum-scale-sc
IBM Spectrum Scale Container Native	ibm-spectrum-scale-sc	ibm-spectrum-scale-sc
Portworx	portworx-fs	portworx-aiops

```
#================================================================================================
===========
# Cloud Pak for AIOps installation variables
#================================================================================================
===========

export CP4AIOPS_NAME=ibm-cp-aiops
export CP4AIOPS_SIZE=large # Set to small if you only require a starter non-production
deployment.
export SECURE_TUNNEL=false # Set to `true` to install Secure Tunnel, otherwise set to `false`.
export PROJECT_CP4AIOPS=cp4aiops
export ACCEPT_LICENSE=false # Set to `true` to agree to the license terms, otherwise install
will fail.
export CATALOG_SRC_CP4AIOPS=ibm-cp-waiops-catalog

# ------------------------------------------------------------------------------------------------
-----------
# Storage
# ------------------------------------------------------------------------------------------------
-----------
export STG_CLASS=<RWX-storage-class-name>
export STG_CLASS_BLOCK=<RWO-storage-class-name>

# ------------------------------------------------------------------------------------------------
-----------
# Your customer details
# ------------------------------------------------------------------------------------------------
-----------
export CUSTOMER_NAME=<your company name>
export CUSTOMER_ICN=<your IBM Customer Number>
export CUSTOMER_ENVIRONMENT=<Set to `trial`, `poc`, or `production`>

# ------------------------------------------------------------------------------------------------
-----------
# `OwnNamespace` installation mode:  leave INSTALL_MODE_NAMESPACE as it is.
# `AllNamespaces` installation mode: change to export INSTALL_MODE_NAMESPACE=openshift-
operators
# ------------------------------------------------------------------------------------------------
-----------
export INSTALL_MODE_NAMESPACE=${PROJECT_CP4AIOPS}

# ------------------------------------------------------------------------------------------------
-----------
# Topology resource group terminology
# Specify `application` or `service` as the terminology to be used for collections of topology
resource
# groups. The default is `application`.
# ------------------------------------------------------------------------------------------------
-----------
export TOPOLOGY_TERMINOLOGY=application
```

If you need help with deciding on the values to set for these environment variables, see the following topics.

- Sizing
- Secure Tunnel
- Operator installation mode

Note: You can set a different value for $PROJECT_CP4AIOPS, $CP4AIOPS_NAME if you want. However, you must not use the **default**, **kube-system**, **kube-public**, **openshift-node**, **openshift-infra**, or **openshift** projects (namespaces) for $PROJECT_CP4AIOPS. This is because IBM Cloud Pak for AIOps uses Security Context Constraints (SCC), and SCCs cannot be assigned to pods created in one of the default Red Hat OpenShift projects (namespaces).

Run the following command to source your script and set the environment variables:

```
. ./waiops_var.sh
```

5.3 Create a custom project (namespace)

Run the following command to create a project (namespace) to deploy IBM Cloud Pak for AIOps into.

```
oc create namespace ${PROJECT_CP4AIOPS}
```

5.4. Configure usage data collection

To help the development of IBM Cloud Pak for AIOps, daily aggregated usage data is collected to analyse how IBM Cloud Pak for AIOps is used. The collection of usage data is enabled by default, but can be disabled. Usage data is collected by the **cp4waiops-metricsprocessor** pod, and stored in the **cp4waiops-metricsprocessor** pod's logs. This usage data is sent to IBM when MustGather output is sent to IBM, as the MustGather includes the output from the **cp4waiops-metricsprocessor** pod's logs. The usage data is then sent to and stored in IBM controlled GDPR-compliant systems. The usage data that is collected is all numeric, and does not include email addresses, passwords, or specific details. Only the following data is collected:

- Current number of applications
- Current number of alerts (all severities aggregated)
- Current number of incidents (all priorities aggregated)
- Current number of policies (includes predefined and user created)
- Current number of runbooks run since installation
- Current number of connectors of each type (For example ServiceNow, Instana, Falcon Logscale)

Configuring the collection of usage data

If you do not want to disable the collection of usage data, run the following command to configure the usage data with your customer details.

```
oc create secret generic aiops-metrics-processor -n ${PROJECT_CP4AIOPS} --from-
literal=customerName=${CUSTOMER_NAME} --from-literal=customerICN=${CUSTOMER_ICN} --from-
literal=environment=${CUSTOMER_ENVIRONMENT}
```

Important: Usage data without your customer details is still collected even if you do not create this secret. If you do not want any usage data collected, then you must run the command given in *Disabling the collection of usage data*.

Disabling the collection of usage data

If you want to disable the collection of usage data, run the following command.

```
oc create secret generic aiops-metrics-processor -n ${PROJECT_CP4AIOPS} --from-
literal=customerName=${CUSTOMER_NAME} --from-literal=customerICN=${CUSTOMER_ICN} --from-
literal=environment=${CUSTOMER_ENVIRONMENT} --from-literal=enableCollection=false
```

Note: You can update your usage data collection preferences after installation. For more information, see Updating usage data collection preferences.

5.5 Verify cluster readiness

Run the following command to switch to your IBM Cloud Pak for AIOps project.

```
oc project ${PROJECT_CP4AIOPS}
```

Run the prerequisite checker script that you downloaded in step 1.1 Download documentation and scripts for offline access to verify whether your Red Hat OpenShift cluster is correctly set up for an IBM Cloud Pak for AIOps installation.

Example output:

```
# ./prereq.sh
Using project "cp4aiops" on server "https://myserver.mycluster.mydomain:6443".

Starting IBM Cloud Pak for AIOps prerequisite checker v4.2...

[INFO] ================================Openshift Container Platform Version
```

```
Check==================================
[INFO] Checking OCP Version. Compatible Versions of OCP is v4.12.x
[INFO] OCP version 4.12.22 is compaitble
[INFO] ==================================Openshift Container Platform Version
Check==================================

[INFO] ==================================Entitlement Pull
Secret==================================
[INFO] Checking whether the Entitlement secret or Global pull secret is configured correctly.
[INFO] Checking if the job 'cp4aiops-entitlement-key-test-job' already exists.
[INFO] The job with name 'cp4aiops-entitlement-key-test-job' was not found, so moving ahead
and creating it.
[INFO] Creating the job 'cp4aiops-entitlement-key-test-job'
job.batch/cp4aiops-entitlement-key-test-job created
[INFO] Verifying if the job 'cp4aiops-entitlement-key-test-job' completed successfully..
[INFO] SUCCESS! Entitlement secret is configured correctly.
job.batch "cp4aiops-entitlement-key-test-job" deleted
[INFO] ==================================Entitlement Pull
Secret==================================

[INFO] ==================================Storage Provider==================================
[INFO] Checking storage providers
[INFO] No IBM Storage Fusion Found... Skipping configuration check.

[INFO] No Portworx StorageClusters found with "Running" or "Online" status. Skipping
configuration check for Portworx.
[INFO] Openshift Data Foundation found.
[INFO] No IBM Cloud Storage found... Skipping configuration check for IBM Cloud Storage Check.

Checking Openshift Data Foundation Configuration...
Verifying if Red Hat Openshift Data Foundation pods are in "Running" or "Completed" status
[INFO] Pods in openshift-storage project are "Running" or "Completed"
[INFO] ocs-storagecluster-ceph-rbd exists.
[INFO] ocs-storagecluster-cephfs exists.
[INFO] No warnings or failures found when checking for Storage Providers.
[INFO] ==================================Storage Provider==================================

[INFO] ==================================Small or Large Profile Install
Resources==================================
[INFO] Checking for cluster resources

[INFO] ==================================Resource
Summary=====================================================
[INFO]                                        Nodes    |      vCPU       |     Memory(GB)
[INFO] Small profile(available/required)   [  14 / 3 ]   [  123 / 62 ]       [  159 / 140 ]
[INFO] Large profile(available/required)   [  14 / 10 ]  [  123 / 162 ]      [  159 / 372 ]
[INFO] ==================================Resource
Summary=====================================================
[INFO] Cluster currently has resources available to create a small profile of Cloud Pak for
AIOps

[INFO] ==================================Small or Large Profile Install
Resources==================================


[INFO] ==================================Prerequisite Checker Tool
Summary==================================
      [  PASS  ] Openshift Container Platform Version Check
      [  PASS  ] Entitlement Pull Secret
      [  PASS  ] Storage Provider
      [  PASS  ] Small or Large Profile Install Resources
[INFO] ==================================Prerequisite Checker Tool
Summary==================================
```

5.6 Install the operator

For more information about installing operators on a cluster, see the Red Hat OpenShift documentation Adding Operators.

For more information about the operators which are installed with IBM Cloud Pak for AIOps, see Operator Details.

1. Create an OperatorGroup.

   Important: Skip this step if you are installing using the 'All Namespaces' installation mode. Check that you set **INSTALL_MODE_NAMESPACE** correctly in step 5.2, and proceed to the next step, Install the IBM Cloud Pak for AIOps operator.

   If you are installing using the 'OwnNamespace' installation mode, then you must create an operator group in your custom project (namespace), or the IBM Cloud Pak for AIOps operator will not install. There might be an operator group for managing a namespace for given APIs. If there is an operator group for the namespace, do not create a second one.

   Create the Operator group by running the following command:

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: cp4aiops-operator-group
  namespace: ${PROJECT_CP4AIOPS}
spec:
  targetNamespaces:
    - "${PROJECT_CP4AIOPS}"
EOF
```

2. Install the IBM Cloud Pak for AIOps operator.

   Run the following command.

```
cat << EOF | oc apply -f -
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-aiops-orchestrator
  namespace: $INSTALL_MODE_NAMESPACE
spec:
  channel: v4.2
  installPlanApproval: Automatic
  name: ibm-aiops-orchestrator
  source: ${CATALOG_SRC_CP4AIOPS}
  sourceNamespace: openshift-marketplace
EOF
```

   Warning: installPlanApproval must not be changed to *Manual*. Manual approval, which requires the manual review and approval of the generated InstallPlans, is not supported. Incorrect timing or ordering of manual approvals of InstallPlans can result in a failed installation.

3. After a few minutes, the IBM Cloud Pak for AIOps operator is installed. Verify that the all of the components have a state of **Succeeded** by running the following command:

```
oc get csv -n ${INSTALL_MODE_NAMESPACE} | egrep "ibm-aiops-orchestrator|ibm-common-service"
```

   Example output:

```
$ oc get csv -n ${INSTALL_MODE_NAMESPACE} | egrep "ibm-aiops-orchestrator|ibm-common-service"
ibm-aiops-orchestrator.v4.2.0         IBM Cloud Pak for AIOps               4.2.0
Succeeded
ibm-common-service-operator.v3.23.7   IBM Cloud Pak foundational services   3.23.7
Succeeded
```

5.7 Create the custom resource

Create an instance of the IBM Cloud Pak for AIOps custom resource. A maximum of one IBM Cloud Pak for AIOps custom resource is allowed per cluster.

1. Use the following YAML to create an instance of the IBM Cloud Pak for AIOps custom resource.

```
cat << EOF | oc apply -f -
apiVersion: orchestrator.aiops.ibm.com/v1alpha1
kind: Installation
metadata:
  name: ${CP4AIOPS_NAME}
  namespace: ${PROJECT_CP4AIOPS}
spec:
  size: ${CP4AIOPS_SIZE}
  storageClass: ${STG_CLASS}
  storageClassLargeBlock: ${STG_CLASS_BLOCK}
  imagePullSecret:
  topologyModel: ${TOPOLOGY_TERMINOLOGY}
  license:
    accept: ${ACCEPT_LICENSE}
  pakModules:
  - name: aiopsFoundation
    enabled: true
  - name: applicationManager
    enabled: true
  - name: aiManager
    enabled: true
  - name: connection
    enabled: ${SECURE_TUNNEL}
EOF
```

Warning: The pakModules **aiopsFoundation**, **applicationManager**, and **aiManager** must be enabled as in the following YAML. Do not change these values to false.

2. Verify your installation.

   Run the following command to check that the **PHASE** of your installation is **Updating**.

   ```
   oc get installations.orchestrator.aiops.ibm.com -n ${PROJECT_CP4AIOPS}
   ```

   Example output:

   ```
   NAME            PHASE      LICENSE    STORAGECLASS    STORAGECLASSLARGEBLOCK    AGE
   ibm-cp-aiops    Updating   Accepted   rook-cephfs     rook-ceph-block           3m
   ```

   It takes around 60-90 minutes for the installation to complete (subject to the speed with which images can be pulled). When installation is complete and successful, the **PHASE** of your installation changes to **Running**. If your installation phase does not change to **Running**, then use the following command to find out which components are not ready:

   ```
   oc get installation.orchestrator.aiops.ibm.com -o yaml -n ${PROJECT_CP4AIOPS} | grep 'Not Ready'
   ```

   Example output:

   ```
   lifecycleservice: Not Ready
   zenservice: Not Ready
   ```

   To see details about why a component is **Not Ready** run the following command, where **<component>** is the component that is not ready, for example **zenservice**.

   ```
   oc get <component> -o yaml -n ${PROJECT_CP4AIOPS}
   ```

   (Optional) If you downloaded the status checker script earlier in step 1.1 Download documentation and scripts for offline access, then you can also run this script to see information about the status of your deployment.

If the installation fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade and Known Issues to help you identify any installation problems.

5.8 Create an EgressFirewall

There is no egress firewall policy defined when you install IBM Cloud Pak for AIOps, so outgoing traffic from workload pods to the internal and external network is unrestricted.

To create a more secure environment, use the following steps.

1. Create an **EgressFirewall** on your Red Hat OpenShift cluster to limit egress from the IBM Cloud Pak for AIOps project (namespace).

   For more information about creating an **EgressFirewall**, see Configuring an egress firewall for a project.

   Note: You can have only one **EgressFirewall** per project/namespace.

2. Configure exceptions to the **EgressFirewall**.

   Edit your **EgressFirewall** to add exceptions for the following IBM Cloud Pak for AIOps components that have egress dependencies, otherwise these IBM Cloud Pak for AIOps components fail when they attempt egress.

   1. Allow egress to any external services, such as the following connections:
      - Kubernetes
      - GitHub
      - Microsoft® Teams
      - ServiceNow
      - Slack
      - VMware® vCenter
   2. Configure your **EgressFirewall** to allow traffic for your GitHub, Kubernetes, ServiceNow, and VMware vCenter integrations.

      Edit your **EgressFirewall** to allow or deny egress, as in the following example:

      ```
      kind: EgressFirewall
      metadata:
        name: default
      spec:
        egress:
        - type: Allow
          to:
            cidrSelector: <1.2.3.0/24>
        - type: Allow
          to:
            dnsName: <www.github.com>
        - type: Allow
          to:
            dnsName: <www.developer.kubernetes.com>
        - type: Allow
          to:
            dnsName: <www.developer.servicenow.com>
        - type: Allow
          to:
            dnsName: <www.developer.vcenter.com>
        - type: Deny
          to:
            cidrSelector: <0.0.0.0/0>
      ```

      Substitute values for `dnsName` and `cidrSelector` that are the DNS names and addresses of your GitHub, Kubernetes, ServiceNow, or VMware vCenter sources.

5.9 Access the Automation console

After you successfully install IBM Cloud Pak for AIOps, get the URL for accessing the Automation console.

1. Use the following command to get the URL to access the Automation console:

   ```
   oc get route -n ${PROJECT_CP4AIOPS} cpd -o jsonpath='{.spec.host}'
   ```

   The following output is a sample output:

   ```
   cpd-cp4aiops.apps.mycluster.mydomain
   ```

   Based on the sample output, your console URL would be **https://cpd-cp4aiops.apps.mycluster.mydomain**

2. Enter the URL in your browser to open the Automation console. Log in with your username and password.

Find the IBM Cloud Pak for AIOps console username and password

The default username to access the Automation console is **admin**. You can check the default username and their password with the following commands.

Note: This information is for the **IBM provided credentials (admin only)** authentication type.

1. Find the default username.

   ```
   oc -n ibm-common-services get secret platform-auth-idp-credentials -o
   jsonpath='{.data.admin_username}' | base64 -d && echo
   ```

2. Get the password for the **_admin_** username.

   ```
   oc -n ibm-common-services get secret platform-auth-idp-credentials -o
   jsonpath='{.data.admin_password}' | base64 -d
   ```

   The following extract shows a sample output:

   ```
   EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA
   ```

   Based on the sample output, your password would be **EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA**.

Important: You can change this default password at any time. For more information, see Changing the cluster administrator password.

What to do next

- Define integrations and applications with Defining.
- If you have an existing on-premises IBM Tivoli Netcool/OMNIbus deployment, then you can connect IBM Cloud Pak for AIOps to it with the Netcool connector. For more information, see Creating IBM Tivoli Netcool/OMNIbus connections.
- If you have an existing on-premises IBM Tivoli Netcool/Impact deployment, then you can connect IBM Cloud Pak for AIOps to it with the IBM Tivoli Netcool/Impact connector. For more information, see Creating IBM Tivoli Netcool/Impact connections.
- Familiarize yourself with backup and restore procedures. It is recommended that you take regular backups of your IBM Cloud Pak for AIOps deployment. For more information, see Backup and restore.

Installing IBM Cloud Pak for AIOps in an air-gapped environment (offline) using a portable device

If your cluster is not connected to the internet, you can deploy an installation of IBM Cloud Pak® for AIOps on your Red Hat® OpenShift® Container Platform cluster by using a portable compute device such as a laptop, or a portable storage device such as a USB device.

In this scenario, your air-gapped (offline) environment has a target registry, and a Red Hat OpenShift cluster on which IBM Cloud Pak for AIOps is to be installed. IBM Cloud Pak for AIOps images are mirrored from the internet to a file system on a portable compute device or a portable storage device. The portable device is then disconnected from the internet and connected in the offline environment, where the images are loaded to the target registry. IBM Cloud Pak for AIOps can then be installed in the offline environment by using the target registry.

Before you begin

- Review the Planning section. Your environment must meet the system requirements.
- If you want to install IBM Cloud Pak for AIOps as a non-root user, you must review the information in Install commands requiring root or sudo.
- Ensure that you are logged in to your Red Hat OpenShift cluster with **oc login** for any steps that use the Red Hat OpenShift command-line interface (CLI).
- If you require details about the permissions that the IBM Cloud Pak for AIOps operators need, see Permissions (IBM Cloud Pak for AIOps).

- Red Hat OpenShift requires a user with `cluster-admin` privileges for the following operations:
  - Create a custom project
  - Configure the cluster
  - Create the catalog source
  - Create an OperatorGroup

Important: The following procedure is based on an Red Hat OpenShift 4.12 environment and includes links for that version. If your environment uses a different supported version of Red Hat OpenShift, ensure that you follow the Red Hat OpenShift documentation for that version.

Installation procedure

1. Set up the mirroring environment
2. Download the CASE
3. Mirror images
4. Configure storage
5. Install IBM Cloud Pak for AIOps

1. Set up the mirroring environment

Prerequisites

Allow access to the following sites and ports:

Table 1. Sites and ports that must be accessible

Site	Description
• `icr.io` • `cp.icr.io` • `dd0.icr.io` • `dd2.icr.io` • `dd4.icr.io` • `dd6.icr.io`	Allow access to these hosts on port 443 to enable access to the `IBM Cloud Container Registry`, CASE OCI artifact, and IBM Cloud Pak® foundational services catalog source.
• `dd1-icr.ibm-zh.com` • `dd3-icr.ibm-zh.com` • `dd5-icr.ibm-zh.com` • `dd7-icr.ibm-zh.com`	If you are located in China, also allow access to these hosts on port 443.
`github.com`	Github houses CASE files, IBM Cloud Pak tools and scripts.
`redhat.com`	Red Hat OpenShift registries that are required for Red Hat OpenShift, and for Red Hat OpenShift upgrades. For more information, see Configuring your firewall for OpenShift Container Platform.

1.1 Download documentation and scripts for offline access

Download the following documentation and scripts that you might need to access during your IBM Cloud Pak for AIOps installation, and copy them to your air-gapped environment.

1. IBM Cloud Pak for AIOps 4.2.0 documentation

   Download the Cloud Pak for AIOps 4.2.0 PDF (this documentation) so that you can access it offline.

2. IBM Cloud Pak for AIOps 4.2.0 scripts

   - The prerequisite checker script verifies whether your Red Hat OpenShift cluster is correctly set up for a IBM Cloud Pak for AIOps installation. You will need to run this script in step 5.5 Verify cluster readiness. Download this script from github.com/IBM.
   - An uninstall script can be downloaded from github.com/IBM.
   - (Optional) The status checker script can be used in step 5.7 Create the custom resource to give information about the status of your deployment. The use of this script is optional, as status can be found directly from the **`ibm-aiops-orchestrator`** custom resource. This script can be downloaded from github.com/IBM.

3. Red Hat OpenShift documentation

   The Red Hat OpenShift documentation can be downloaded for offline access from Red Hat. The **`Security and compliance`**, **`Installing`**, **`CLI Tools`**, **`Images`**, and **`Operators`** sections are referenced by this documentation.

1.2 Install and configure Red Hat OpenShift

IBM Cloud Pak for AIOps requires Red Hat OpenShift to be installed and running on your target cluster. You must have administrative access to your Red Hat OpenShift cluster.

For more information about the supported versions of Red Hat OpenShift, see Supported Red Hat OpenShift Container Platform versions. The hardware architecture that you install IBM Cloud Pak for AIOps on must be AMD64.

1. Install Red Hat OpenShift by using the instructions in the Red Hat OpenShift documentation. Information about installing a cluster in a restricted network is given in Mirroring images for a disconnected installation.

2. Install the Red Hat OpenShift command line interface (**`oc`**) on your cluster's boot node and run **`oc login`**. For more information, see the instructions in Getting started with the Red Hat OpenShift CLI.

3. Ensure that the clocks on your Red Hat OpenShift cluster are synchronized. Each Red Hat OpenShift node in the cluster must have access to an NTP server. Red Hat OpenShift nodes use NTP to synchronize their clocks. IBM Cloud Pak for AIOps runs on Red Hat OpenShift and also has this requirement. Discrepancies between the clocks on the Red Hat OpenShift nodes can cause IBM Cloud Pak for AIOps to experience operational issues. See the Red Hat OpenShift documentation for information about how to use a **`MachineConfig`** custom resource to configure chrony to connect to your NTP servers.

4. Optionally configure a custom certificate for IBM Cloud Pak for AIOps to use. You can use either of the following methods:

   - Configure a custom certificate for the Red Hat OpenShift cluster. Follow the instructions in the Red Hat OpenShift documentation Replacing the default ingress certificate. Then, deploy the signing CA certificate into the cluster by following the instructions in the Red Hat OpenShift documentation Replacing the CA Bundle certificate.
   - If you would like to use a custom certificate for IBM Cloud Pak for AIOps only, then after installation is complete follow the instructions in Using a custom certificate.

1.3 Set up a target registry

You must have a local Docker type production-grade registry available in the air-gapped environment to store the IBM Cloud Pak for AIOps images in. The registry must meet the following requirements:

- supports Docker Manifest V2, Schema 2.
- supports multi-architecture images.
- is accessible from the Red Hat OpenShift cluster nodes.
- allows path separators in the image name.
- you have the username and password for a user who can read from and write to the registry.
- must have 148 GB of storage to hold all of the software that is to be transferred to the target registry.

If you do not already have a suitable production-grade registry available, then you must install and configure one. For more information, see About the mirror registry in the Red Hat OpenShift documentation.

Important: Do not use the Red Hat OpenShift image registry as your target registry. The Red Hat OpenShift registry does not support multi-architecture images or path separators in the image name.

1.4 Prepare a host

Prepare a portable compute device, or if you are using a portable storage device then a connected compute device, as follows.

You must be able to connect your host to the internet. Your host must be on a Linux® x86_64 or Mac platform with any operating system that the IBM Cloud Pak CLI and the Red Hat OpenShift CLI support. If you are on a Windows® platform, you must run the actions in a Linux® x86_64 VM or from a Windows Subsystem for Linux® (WSL) terminal.

Your portable device and any intermediary devices must have 148 GB of storage to hold all the software that is to be transferred to the local target registry.

Complete the following steps on your host.

1. Install Docker or Podman. One of these tools is needed for container management.

   To install Podman, see the Podman Installation Instructions.

   To install Docker (for example, on Red Hat® Enterprise Linux®), run the following commands.

   ```
   yum check-update
   yum install docker
   ```

   Note: Docker is not shipped or supported for Red Hat Enterprise Linux (RHEL) 8. The Podman container engine replaced docker as the preferred, maintained, and supported container runtime of choice for Red Hat Enterprise Linux 8 systems. For more information, see Running containers without Docker in the Red Hat documentation.

2. Install the Red Hat OpenShift CLI tool, **`oc`**.

   `oc` is required for Red Hat OpenShift management. For more information, see Getting started with the Red Hat OpenShift CLI in the Red Hat OpenShift documentation.

1.5 Install the IBM Catalog Management Plug-in for IBM Cloud Pak®

The IBM Catalog Management Plug-in for IBM Cloud Pak (**`ibm-pak-plugin`**) is used for the deployment of IBM Cloud Paks® in a disconnected environment. It simplifies the process for discovering required IBM product images and uses standard tools for registry and cluster access. The **`ibm-pak-plugin`** also extends the Red Hat OpenShift CLI (**`oc`**) capability to streamline the process of delivering installation images to the IBM Cloud Pak in an air-gapped environment.

1. Download and install the most recent version of the **`IBM Catalog Management Plug-in for IBM Cloud Paks`** for your host operating system from github.com/IBM.

2. Run the following command to extract the files.

   ```
   tar -xf oc-ibm_pak-linux-amd64.tar.gz
   ```

3. Run the following command to move the file to the **`/usr/local/bin`** directory.

   ```
   mv oc-ibm_pak-linux-amd64 /usr/local/bin/oc-ibm_pak
   ```

   Note: If you are installing as a non-root user, you must use sudo.

4. Confirm that **`ibm-pak-plugin`** is installed by running the following command.

   ```
   oc ibm-pak --help
   ```

   Expected result: The **`ibm-pak-plugin`** usage is displayed.

2. Download the CASE

Set environment variables on the portable device, and connect it to the internet so that you can download the IBM Cloud Pak for AIOps CASE files.

Note: Save a copy of your environment variable values to a file that you can use as a reference when you are completing your air-gapped installation tasks.

1. Create the following environment variables.

```
export CASE_NAME=ibm-cp-waiops
export CASE_VERSION=1.8.0
export CASE_INVENTORY_SETUP=cpwaiopsSetup
export TARGET_REGISTRY_HOST=<IP_or_FQDN_of_target_registry>
export TARGET_REGISTRY_PORT=<port_number_of_target_registry>
export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
export TARGET_REGISTRY_USER=<username>
export TARGET_REGISTRY_PASSWORD=<password>
```

   The target registry is the registry where the IBM Cloud Pak for AIOps images are mirrored to, and accessed from by the Red Hat OpenShift cluster, as setup in 1.3 Set up a target registry

   If your portable device must connect to the internet through a proxy, then also set the following environment variables.

```
export https_proxy=http://proxy-server-hostname:port
export http_proxy=http://proxy-server-hostname:port
```

2. Connect your portable device to the internet and disconnect it from the air-gapped environment.

3. Download the IBM Cloud Pak for AIOps installer and image inventory to your portable device.

```
oc ibm-pak get ${CASE_NAME} --version ${CASE_VERSION}
```

   The CASE is downloaded to **~/.ibm-pak/data/cases/$CASE_NAME/$CASE_VERSION**. The logs files are available at ~/.ibm-pak/logs/oc-ibm_pak.log.

   Note: If you do not specify the CASE version, then the latest CASE is downloaded. The root directory that is used by the **ibm-pak-plugin** is **~/.ibm-pak**. If required, the root directory can be configured by setting the **IBMPAK_HOME** environment variable.

3. Mirror images

Complete the following steps to mirror the IBM Cloud Pak for AIOps images from the internet to the file system on your portable device, and then from the file system to the target registry in the air-gapped environment.

3.1. Generate mirror manifests

Run the following command to generate mirror manifests to be used when mirroring the images to the target registry.

```
oc ibm-pak generate mirror-manifests ${CASE_NAME} file://cpfs --version ${CASE_VERSION} --
final-registry ${TARGET_REGISTRY}/cpfs
```

Table 2. Parameter description

Argument	Description
file://cpfs	Specifies the path extension where images are mirrored to. Images are mirrored to **$IMAGE_PATH/cpfs** when the **oc image mirror** command is run with **images-mapping-to-filesystem.txt**. For more information, see Mirror the images to the file system.
$TARGET_REGISTRY/cpfs	This argument generates an image-mapping file that is used by the **oc image mirror** commands to mirror images to the **TARGET_REGISTRY** at namespace **cpfs**. For example, if the CASE you are installing needs the image, **quay.io/opencloudio/ibm-events-kafka-2.6.0@sha256:10d422dddd29ff19c87066fc6510eee05f5fa4ff608b87a06e898b3b6a3a13c7**, then its final URL in your target registry will be **$TARGET_REGISTRY/cpfs/opencloudio/ibm-events-kafka-2.6.0**. Note the new namespace of **cpfs** in the URL. The namespace path can be multi level if your target registry supports it.

The command generates the following files at **~/.ibm-pak/data/mirror/$CASE_NAME/$CASE_VERSION**:

- images-mapping-to-filesystem.txt
- images-mapping-from-filesystem.txt
- image-content-source-policy.yaml

3.2. Authenticate with the IBM Entitled Registry

Log in to the IBM Entitled Registry to generate an authentication file containing the IBM Entitled Registry credentials, and then create an environment variable that has the location of the authentication file. This file is used later to enable the **`oc image mirror`** command to pull the images from the IBM Entitled Registry.

1. Get the authentication credentials for the IBM Entitled Registry.

    1. To obtain the entitlement key that is assigned to your IBMid, log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.

    2. In the *Entitlement keys* section, select **Copy key** to copy the entitlement key.

2. Run the following command to create an environment variable that contains your entitlement key.

    ```
    export ENTITLED_REGISTRY_PASSWORD=<key>
    ```

    Where **`<key>`** is the entitlement key that you copied in the previous step.

3. Store the authentication credentials for the IBM Entitled Registry.

    If you are using Podman, run the following commands:

    ```
    podman login cp.icr.io -u cp -p ${ENTITLED_REGISTRY_PASSWORD}
    export REGISTRY_AUTH_FILE=${XDG_RUNTIME_DIR}/containers/auth.json
    unset ENTITLED_REGISTRY_PASSWORD
    ```

    Note: The authentication file is usually at **`${XDG_RUNTIME_DIR}/containers/auth.json`**. For more information, see the Options section in the Podman documentation.

    If you are using Docker, run the following commands:

    ```
    docker login cp.icr.io -u cp -p ${ENTITLED_REGISTRY_PASSWORD}
    export REGISTRY_AUTH_FILE=$HOME/.docker/config.json
    unset ENTITLED_REGISTRY_PASSWORD
    ```

    Note: The authentication file is usually at **`$HOME/.docker/config.json`**. For more information, see the Docker documentation.

3.3. Mirror the images to the file system

Complete these steps to mirror the images from the internet to a file system on your portable device.

1. Create an environment variable to store the location of the file system where the images are to be stored.

    ```
    export IMAGE_PATH=<image-path>
    ```

    Where **`<image-path>`** is the directory where you want the images to be stored.

2. Run the following command to mirror the images from the IBM Entitled Registry to the file system.

    ```
    nohup oc image mirror \
    -f ~/.ibm-pak/data/mirror/${CASE_NAME}/${CASE_VERSION}/images-mapping-to-filesystem.txt \
    -a ${REGISTRY_AUTH_FILE} \
    --filter-by-os '.*' \
    --insecure \
    --skip-multiple-scopes \
    --dir "${IMAGE_PATH}" \
    --max-per-registry=1 > my-mirror-progress.txt 2>&1 &
    ```

    The UNIX® command **`nohup`** is used to ensure that the mirroring process continues even if there is a loss of network connection, and redirection of output to a file provides improved monitoring and error visibility.

    Run the following command if you want to see the progress of the mirroring:

    ```
    tail -f my-mirror-progress.txt
    ```

    Note: If an error occurs during mirroring, the mirror command can be rerun.

3.4 Setup the file system in the air-gapped environment

1. Copy files to the air-gapped environment (portable storage device only)

   If you are using a portable storage device, you must copy the files from the portable storage device to a local compute device in the air-gapped environment that has access to the target registry. If you are using a portable compute device, then these items are already present, and you can proceed to the next step.

   Copy the following items to your local compute device:

   - the file system located at $IMAGE_PATH, which you specified earlier
   - `~/.ibm-pak` directory
2. Disconnect the device that has your file system, (the portable compute device or the local compute device) from the internet and connect it to the air-gapped environment.

3. Ensure that environment variables are set on the device in the air-gapped environment that has access to the target registry.

   If you are using a portable storage device, then set the following environment variables on your local compute device within the air-gapped environment.

   If you are using a portable compute device that you have restarted since mirroring the images, then your environment variables will have been lost and you will need to set the following environment variables on your portable compute device again.

```
export CASE_NAME=ibm-cp-waiops
export CASE_VERSION=1.8.0
export CASE_INVENTORY_SETUP=cpwaiopsSetup
export TARGET_REGISTRY_HOST=<IP_or_FQDN_of_target_registry>
export TARGET_REGISTRY_PORT=<port_number_of_target_registry>
export TARGET_REGISTRY=${TARGET_REGISTRY_HOST}:${TARGET_REGISTRY_PORT}
export TARGET_REGISTRY_USER=<username>
export TARGET_REGISTRY_PASSWORD=<password>
export IMAGE_PATH=<image_path>
```

3.5 Authenticate with the target registry

Authenticate with the target registry in the air-gapped environment that you are mirroring the images into.

If you are using Podman, run the following command:

```
podman login ${TARGET_REGISTRY} -u ${TARGET_REGISTRY_USER} -p ${TARGET_REGISTRY_PASSWORD}
export REGISTRY_AUTH_FILE=${XDG_RUNTIME_DIR}/containers/auth.json
```

Note: The authentication file is usually at `${XDG_RUNTIME_DIR}/containers/auth.json`. For more information, see the Options section in the Podman documentation.

If you are using Docker, run the following command:

```
docker login ${TARGET_REGISTRY} -u ${TARGET_REGISTRY_USER} -p ${TARGET_REGISTRY_PASSWORD}
export REGISTRY_AUTH_FILE=$HOME/.docker/config.json
```

Note: The authentication file is usually at `$HOME/.docker/config.json`. For more information, see the Docker documentation.

3.6 Mirror the images to the target registry from the file system

Complete the steps in this section on the device that has your file system (the portable compute device or the local compute device) to copy the images from the file system to the $TARGET_REGISTRY. Your device with the file system must be connected to both the target registry and the Red Hat OpenShift cluster.

Run the following command to copy the images referenced in the `images-mapping-from-filesystem.txt` from the $IMAGE_PATH file system to the final target registry.

```
nohup oc image mirror \
-f ~/.ibm-pak/data/mirror/${CASE_NAME}/${CASE_VERSION}/images-mapping-from-filesystem.txt \
--from-dir "${IMAGE_PATH}" \
-a ${REGISTRY_AUTH_FILE} \
```

```
--filter-by-os '.*' \
--insecure  \
--skip-multiple-scopes \
--max-per-registry=1 > my-mirror-progress2.txt 2>&1 &
```

The UNIX command **nohup** is used to ensure that the mirroring process continues even if there is a loss of network connection, and redirection of output to a file provides improved monitoring and error visibility.

Run the following command if you want to see the progress of the mirroring:

```
tail -f my-mirror-progress2.txt
```

Note: If an error occurs during mirroring, the mirror command can be rerun.

3.7 Configure the cluster

1. Log in to your Red Hat OpenShift cluster.

   You can identify your specific **oc login** command by clicking the user menu in the upper left of the Red Hat OpenShift console, and then clicking **Copy Login Command**.

   Example:

   ```
   oc login <server> -u <cluster username> -p <cluster pass>
   ```

2. Update the global image pull secret for your Red Hat OpenShift cluster.

   Follow the steps in the Red Hat OpenShift documentation topic Updating the global cluster pull secret.

   These steps enable your cluster to have authentication credentials in place to pull images from your TARGET_REGISTRY as specified in the **image-content-source-policy.yaml**, which you will apply to your cluster in the next step.

3. Create the **ImageContentSourcePolicy**.

   Run the following command to create an ImageContentSourcePolicy (ICSP).

   ```
   oc apply -f  ~/.ibm-pak/data/mirror/${CASE_NAME}/${CASE_VERSION}/image-content-source-policy.yaml
   ```

4. Verify that the **ImageContentSourcePolicy** resource is created.

   ```
   oc get imagecontentsourcepolicy
   ```

   Example output, showing a newly created **ibm-cp-waiops** ICSP.

   ```
   oc get imagecontentsourcepolicy
   NAME                        AGE
   ibm-cp-waiops               95s
   redhat-operator-index-0     5d18h
   ```

5. Verify your cluster node status.

   ```
   oc get MachineConfigPool -w
   ```

   Important: After the **ImageContentsourcePolicy** and global image pull secret are applied, the configuration of your nodes will be updated sequentially. Wait until all the **MachineConfigPools** are updated before proceeding to the next step.

6. (Optional) If you use an insecure registry, you must add the target registry to the cluster's **insecureRegistries** list.

   ```
   oc patch image.config.openshift.io/cluster --type=merge \
   -p '{"spec":{"registrySources":{"insecureRegistries":["'${TARGET_REGISTRY}'"]}}}'
   ```

4. Configure storage

The storage configuration must satisfy your sizing requirements. Two storage classes are needed for installing IBM Cloud Pak for AIOps. For more information, see Storage.

5. Install IBM Cloud Pak for AIOps

Now that the images are mirrored to your air-gapped environment, you can deploy IBM Cloud Pak for AIOps to that environment. To install IBM Cloud Pak for AIOps, complete the following steps.

- 5.1 Create the catalog source
- 5.2 Create environment variables
- 5.3 Create a custom project
- 5.4 Configure usage data collection
- 5.5 Verify cluster readiness
- 5.6 Install the operator
- 5.7 Create the custom resource
- 5.8 Create an EgressFirewall
- 5.9 Access the Automation console

5.1 Create the catalog source

Now that the images are mirrored to your air-gapped environment, you can deploy IBM Cloud Pak for AIOps to that environment

1. Create and configure the catalog source.

```
oc ibm-pak launch \
${CASE_NAME} \
--version ${CASE_VERSION} \
--action install-catalog \
--inventory ${CASE_INVENTORY_SETUP} \
--namespace openshift-marketplace \
--args "--registry ${TARGET_REGISTRY}/cpfs --recursive \
--inputDir ~/.ibm-pak/data/cases/${CASE_NAME}/${CASE_VERSION}"
```

2. Verify that the CatalogSource is installed.

```
oc get catalogsource -n openshift-marketplace
oc get pods -n openshift-marketplace
```

Example output:

```
oc get catalogsource -n openshift-marketplace
NAME                                          DISPLAY                                          TYPE
PUBLISHER   AGE
aiops-analytics-operator-catalog              IBM AIOps Analytics Product Catalog              grpc
IBM         25m
aiops-lifecycle-operator-catalog              IBM AIOps Lifecycle Product Catalog              grpc
IBM         25m
certified-operators                           Certified Operators                              grpc
Red Hat     24h
cloud-native-postgresql-catalog               Cloud Native Postgresql Catalog                  grpc
IBM         54m
community-operators                           Community Operators                              grpc
Red Hat     24h
ibm-aiops-ir-core-operators                   IBM AIOps Issue Resolution Core Operators        grpc
IBM         54m
ibm-aiopsedge-case                            aiopsedge index                                  grpc
IBM         25m
ibm-asm-catalog                               IBM Agile Service Manager Catalog                grpc
IBM         25m
ibm-cloud-databases-redis-operator-catalog    ibm-cloud-databases-redis-operator-catalog       grpc
IBM         25m
ibm-cp-waiops-catalog                         IBM Cloud Pak for AIOps Catalog                  grpc
IBM         25m
ibm-cp-waiops-elastic-catalog                 IBM CP4AIOps - Elastic                           grpc
IBM         25m
ibm-cp-waiops-flink-catalog                   IBM CP4AIOps - Flink                             grpc
IBM         25m
ibm-kong-management-catalog                   IBM Kong Management Catalog                      grpc
IBM         25m
ibm-secure-tunnel-operator-catalog            IBM Secure Tunnel operator Catalog               grpc
IBM         25m
```

```
ibm-watson-ai-manager-catalog                ibm-watson-ai-manager-catalog                grpc
IBM         25m
ibm-watson-aiops-ui-catalog                  IBM WAIOps UI Catalog                        grpc
IBM         25m
opencloud-operators                          IBMCS Operators                              grpc
IBM         25m
redhat-marketplace                           Red Hat Marketplace                          grpc
Red Hat     24h
redhat-operators                             Red Hat Operators                            grpc
Red Hat     24h
<br>
oc get pods -n openshift-marketplace
NAME                                                              READY   STATUS      RESTARTS
AGE
0935b4423e67bb338b8687eeee2898674099d8fc158ffb50b8bc1fba0a692g2   0/1     Completed   0
24h
1a64439d8a73c4c057540c5e6f4329f15aeaa15b5cc9f254df9ab9dd7bjqzpt   0/1     Completed   0
24h
6766646cfd55d8f1be544bc968431245406bd8d357bcc6c75efb34bb50bc829   0/1     Completed   0
24h
aiops-analytics-operator-catalog-cvwcc                            1/1     Running     0
3m45s
aiops-lifecycle-operator-catalog-2kv2l                            1/1     Running     0
3m46s
certified-operators-vn8ql                                         1/1     Running     0
86m
cloud-native-postgresql-catalog-m6c42                             1/1     Running     0
3m32s
community-operators-h59qg                                         1/1     Running     0
65m
e370676c5b33f2c041ba1d3a7bdc078b247d7d4e6fd61841e5d197770e9n2sf   0/1     Completed   0
24h
f8ea7cf3f9611828a4809f2eaaf2ae944796f25e2ea7c23bb389c7509cpjxf7   0/1     Completed   0
24h
ibm-aiops-ir-core-operators-k69f9                                 1/1     Running     0
3m49s
ibm-aiopsedge-case-wl2nx                                          1/1     Running     0
3m41s
ibm-asm-catalog-qgz4l                                             1/1     Running     0
3m51s
ibm-cloud-databases-redis-operator-catalog-tp7jd                  1/1     Running     0
3m35s
ibm-cp-waiops-catalog-khjmt                                       1/1     Running     0
3m32s
ibm-cp-waiops-elastic-catalog-ptv77                               1/1     Running     0
3m30s
ibm-cp-waiops-flink-catalog-fhs69                                 1/1     Running     0
3m29s
ibm-kong-management-catalog-mr89f                                 1/1     Running     0
3m56s
ibm-secure-tunnel-operator-catalog-76fvf                          1/1     Running     0
3m39s
ibm-watson-ai-manager-catalog-bz66v                               1/1     Running     0
3m53s
ibm-watson-aiops-ui-catalog-9srhm                                 1/1     Running     0
3m55s
marketplace-operator-7dd7c98f68-2t5r9                             1/1     Running     0
25h
opencloud-operators-79j9k                                         1/1     Running     0
3m43s
redhat-marketplace-wc674                                          1/1     Running     0
24h
redhat-operators-rc4b2                                            1/1     Running     0
6h55m
```

5.2 Create environment variables

Create and then source a shell script named *waiops_var.sh* that defines the environment variables which will be used to provide installation parameters for your deployment. Use the following codeblock as a template, replacing the brackets **< ... >** with values for your environment.

You can use the following table to find the values to set for your storage environment variables.

Table 3. Storage provider classes

Storage provider	RWX-storage-class-name	RWO-storage-class-name
IBM Cloud®	ibmc-file-gold-gid	ibmc-block-gold
Red Hat® OpenShift® Data Foundation	ocs-storagecluster-cephfs	ocs-storagecluster-ceph-rbd
IBM Storage Fusion	ibm-spectrum-scale-sc	ibm-spectrum-scale-sc
IBM Spectrum Scale Container Native	ibm-spectrum-scale-sc	ibm-spectrum-scale-sc
Portworx	portworx-fs	portworx-aiops

```
#============================================================================================================
# Cloud Pak for AIOps installation variables
#============================================================================================================

export CP4AIOPS_NAME=ibm-cp-aiops
export CP4AIOPS_SIZE=large # Set to small if you only require a starter non-production
deployment.
export SECURE_TUNNEL=false # Set to `true` to install Secure Tunnel, otherwise set to `false`.
export PROJECT_CP4AIOPS=cp4aiops
export ACCEPT_LICENSE=false # Set to `true` to agree to the license terms, otherwise install
will fail.
export CATALOG_SRC_CP4AIOPS=ibm-cp-waiops-catalog

# ------------------------------------------------------------------------------------------------
# Storage
# ------------------------------------------------------------------------------------------------
export STG_CLASS=<RWX-storage-class-name>
export STG_CLASS_BLOCK=<RWO-storage-class-name>

# ------------------------------------------------------------------------------------------------
# Your customer details
# ------------------------------------------------------------------------------------------------
export CUSTOMER_NAME=<your company name>
export CUSTOMER_ICN=<your IBM Customer Number>
export CUSTOMER_ENVIRONMENT=<Set to `trial`, `poc`, or `production`>

# ------------------------------------------------------------------------------------------------
# `OwnNamespace` installation mode:  leave INSTALL_MODE_NAMESPACE as it is.
# `AllNamespaces` installation mode: change to export INSTALL_MODE_NAMESPACE=openshift-
operators
# ------------------------------------------------------------------------------------------------
export INSTALL_MODE_NAMESPACE=${PROJECT_CP4AIOPS}

# ------------------------------------------------------------------------------------------------
# Topology resource group terminology
# Specify `application` or `service` as the terminology to be used for collections of topology
resource
# groups. The default is `application`.
# ------------------------------------------------------------------------------------------------
export TOPOLOGY_TERMINOLOGY=application
```

If you need help with deciding on the values to set for these environment variables, see the following topics.

- Sizing
- Secure Tunnel
- Operator installation mode

Note: You can set a different value for $PROJECT_CP4AIOPS and $CP4AIOPS_NAME if you want. However, you must not use the **default**, **kube-system**, **kube-public**, **openshift-node**, **openshift-infra**, or **openshift** projects (namespaces) for $PROJECT_CP4AIOPS. This is because IBM Cloud Pak for AIOps uses Security Context Constraints (SCC), and SCCs cannot be assigned to pods created in one of the default Red Hat OpenShift projects (namespaces).

Run the following command to source your script and set the environment variables:

```
. ./waiops_var.sh
```

5.3 Create a custom project (namespace)

Run the following command to create a project (namespace) to deploy IBM Cloud Pak for AIOps into.

```
oc create namespace ${PROJECT_CP4AIOPS}
```

5.4. Configure usage data collection

To help the development of IBM Cloud Pak for AIOps, daily aggregated usage data is collected to analyse how IBM Cloud Pak for AIOps is used. The collection of usage data is enabled by default, but can be disabled. Usage data is collected by the **cp4waiops-metricsprocessor** pod, and stored in the **cp4waiops-metricsprocessor** pod's logs. This usage data is sent to IBM when MustGather output is sent to IBM, as the MustGather includes the output from the **cp4waiops-metricsprocessor** pod's logs. The usage data is then sent to and stored in IBM controlled GDPR-compliant systems. The usage data that is collected is all numeric, and does not include email addresses, passwords, or specific details. Only the following data is collected:

- Current number of applications
- Current number of alerts (all severities aggregated)
- Current number of incidents (all priorities aggregated)
- Current number of policies (includes predefined and user created)
- Current number of runbooks run since installation
- Current number of connectors of each type (For example ServiceNow, Instana, Falcon Logscale)

This usage data is sent to IBM only if the MustGather tool is run and sent to IBM. The MustGather includes the output from the **cp4waiops-metricsprocessor** pod's logs.

Configuring the collection of usage data

If you do not want to disable the collection of usage data, run the following command to configure the usage data with your customer details.

```
oc create secret generic aiops-metrics-processor -n ${PROJECT_CP4AIOPS} --from-
literal=customerName=${CUSTOMER_NAME} --from-literal=customerICN=${CUSTOMER_ICN} --from-
literal=environment=${CUSTOMER_ENVIRONMENT}
```

Important: Usage data without your customer details is still collected even if you do not create this secret. If you do not want any usage data collected, then you must run the command given in *Disabling the collection of usage data*.

Disabling the collection of usage data

If you want to disable the collection of usage data, run the following command.

```
oc create secret generic aiops-metrics-processor -n ${PROJECT_CP4AIOPS} --from-
literal=customerName=${CUSTOMER_NAME} --from-literal=customerICN=${CUSTOMER_ICN} --from-
literal=environment=${CUSTOMER_ENVIRONMENT} --from-literal=enableCollection=false
```

Note: You can update your usage data collection preferences after installation. For more information, see Updating usage data collection preferences.

5.5 Verify cluster readiness

Run the following command to switch to your IBM Cloud Pak for AIOps project.

```
oc project ${PROJECT_CP4AIOPS}
```

Run the prerequisite checker script that you downloaded in step 1.1 Download documentation and scripts for offline access to verify whether your Red Hat OpenShift cluster is correctly set up for an IBM Cloud Pak for AIOps installation.

Example output:

```
# ./prereq.sh
Using project "cp4aiops" on server "https://myserver.mycluster.mydomain:6443".

Starting IBM Cloud Pak for AIOps prerequisite checker v4.2...

[INFO] ================================Openshift Container Platform Version
Check================================
[INFO] Checking OCP Version. Compatible Versions of OCP is v4.12.x
[INFO] OCP version 4.12.22 is compaitble
[INFO] ================================Openshift Container Platform Version
Check================================

[INFO] ================================Entitlement Pull
Secret================================
[INFO] Checking whether the Entitlement secret or Global pull secret is configured correctly.
[INFO] Checking if the job 'cp4aiops-entitlement-key-test-job' already exists.
[INFO] The job with name 'cp4aiops-entitlement-key-test-job' was not found, so moving ahead
and creating it.
[INFO] Creating the job 'cp4aiops-entitlement-key-test-job'
job.batch/cp4aiops-entitlement-key-test-job created
[INFO] Verifying if the job 'cp4aiops-entitlement-key-test-job' completed successfully..
[INFO] SUCCESS! Entitlement secret is configured correctly.
job.batch "cp4aiops-entitlement-key-test-job" deleted
[INFO] ================================Entitlement Pull
Secret================================

[INFO] ================================Storage Provider================================
[INFO] Checking storage providers
[INFO] No IBM Storage Fusion Found... Skipping configuration check.

[INFO] No Portworx StorageClusters found with "Running" or "Online" status. Skipping
configuration check for Portworx.
[INFO] Openshift Data Foundation found.
[INFO] No IBM Cloud Storage found... Skipping configuration check for IBM Cloud Storage Check.

Checking Openshift Data Foundation Configuration...
Verifying if Red Hat Openshift Data Foundation pods are in "Running" or "Completed" status
[INFO] Pods in openshift-storage project are "Running" or "Completed"
[INFO] ocs-storagecluster-ceph-rbd exists.
[INFO] ocs-storagecluster-cephfs exists.
[INFO] No warnings or failures found when checking for Storage Providers.
[INFO] ================================Storage Provider================================

[INFO] ================================Small or Large Profile Install
Resources================================
[INFO] Checking for cluster resources

[INFO] ================================Resource
Summary=====================================================
[INFO]                                        Nodes   |      vCPU      |     Memory(GB)
[INFO] Small profile(available/required)  [  14 / 3 ]   [  123 / 62 ]       [  159 / 140 ]
[INFO] Large profile(available/required)  [  14 / 10 ]  [  123 / 162 ]      [  159 / 372 ]
[INFO] ================================Resource
Summary=====================================================
[INFO] Cluster currently has resources available to create a small profile of Cloud Pak for
AIOps

[INFO] ================================Small or Large Profile Install
Resources================================


[INFO] ================================Prerequisite Checker Tool
Summary================================
      [  PASS  ] Openshift Container Platform Version Check
      [  PASS  ] Entitlement Pull Secret
      [  PASS  ] Storage Provider
      [  PASS  ] Small or Large Profile Install Resources
[INFO] ================================Prerequisite Checker Tool
Summary================================
```

5.6 Install the operator

For more information about installing operators on a cluster, see the Red Hat OpenShift documentation Adding Operators.

For more information about the operators which are installed with IBM Cloud Pak for AIOps, see Operator Details.

1. Create an OperatorGroup.

   Important: Skip this step if you are installing using the 'All Namespaces' installation mode. Check that you set **INSTALL_MODE_NAMESPACE** correctly in step 5.2, and proceed to the next step, Install the IBM Cloud Pak for AIOps operator.

   If you are installing using the 'OwnNamespace' installation mode, then you must create an operator group in your custom project (namespace), or the IBM Cloud Pak for AIOps operator will not install. There might be an operator group for managing a namespace for given APIs. If there is an operator group for the namespace, do not create a second one.

   Create the Operator group by running the following command:

   ```
   cat << EOF | oc apply -f -
   apiVersion: operators.coreos.com/v1
   kind: OperatorGroup
   metadata:
     name: cp4aiops-operator-group
     namespace: ${PROJECT_CP4AIOPS}
   spec:
     targetNamespaces:
       - "${PROJECT_CP4AIOPS}"
   EOF
   ```

2. Install the IBM Cloud Pak for AIOps operator.

   Run the following command.

   ```
   cat << EOF | oc apply -f -
   apiVersion: operators.coreos.com/v1alpha1
   kind: Subscription
   metadata:
     name: ibm-aiops-orchestrator
     namespace: $INSTALL_MODE_NAMESPACE
   spec:
     channel: v4.2
     installPlanApproval: Automatic
     name: ibm-aiops-orchestrator
     source: ${CATALOG_SRC_CP4AIOPS}
     sourceNamespace: openshift-marketplace
   EOF
   ```

   Warning: installPlanApproval must not be changed to *Manual*. Manual approval, which requires the manual review and approval of the generated InstallPlans, is not supported. Incorrect timing or ordering of manual approvals of InstallPlans can result in a failed installation.

3. After a few minutes, the IBM Cloud Pak for AIOps operator is installed. Verify that the all of the components have a state of **Succeeded** by running the following command:

   ```
   oc get csv -n ${INSTALL_MODE_NAMESPACE} | egrep "ibm-aiops-orchestrator|ibm-common-service"
   ```

   Example output:

   ```
   $ oc get csv -n ${INSTALL_MODE_NAMESPACE} | egrep "ibm-aiops-orchestrator|ibm-common-service"
   ibm-aiops-orchestrator.v4.2.0          IBM Cloud Pak for AIOps                4.2.0    Succeeded
   ibm-common-service-operator.v3.23.7    IBM Cloud Pak foundational services    3.23.7   Succeeded
   ```

5.7 Create the custom resource

Create an instance of the IBM Cloud Pak for AIOps custom resource. A maximum of one IBM Cloud Pak for AIOps custom resource is allowed per cluster.

1. Use the following YAML to create an instance of the IBM Cloud Pak for AIOps custom resource.

```
cat << EOF | oc apply -f -
apiVersion: orchestrator.aiops.ibm.com/v1alpha1
kind: Installation
metadata:
  name: ${CP4AIOPS_NAME}
  namespace: ${PROJECT_CP4AIOPS}
spec:
  size: ${CP4AIOPS_SIZE}
  storageClass: ${STG_CLASS}
  storageClassLargeBlock: ${STG_CLASS_BLOCK}
  imagePullSecret:
  topologyModel: ${TOPOLOGY_TERMINOLOGY}
  license:
    accept: ${ACCEPT_LICENSE}
  pakModules:
  - name: aiopsFoundation
    enabled: true
  - name: applicationManager
    enabled: true
  - name: aiManager
    enabled: true
  - name: connection
    enabled: ${SECURE_TUNNEL}
EOF
```

   Warning: The pakModules **aiopsFoundation**, **applicationManager**, and **aiManager** must be enabled as in the following YAML. Do not change these values to false.

2. Verify your installation.

   Run the following command to check that the **PHASE** of your installation is **Updating**.

```
oc get installations.orchestrator.aiops.ibm.com -n ${PROJECT_CP4AIOPS}
```

   Example output:

```
NAME                    PHASE      LICENSE    STORAGECLASS    STORAGECLASSLARGEBLOCK    AGE
ibm-cp-aiops            Updating   Accepted   rook-cephfs     rook-ceph-block           3m
```

   It takes around 60-90 minutes for the installation to complete (subject to the speed with which images can be pulled). When installation is complete and successful, the **PHASE** of your installation changes to **Running**. If your installation phase does not change to **Running**, then use the following command to find out which components are not ready:

```
oc get installation.orchestrator.aiops.ibm.com -o yaml -n ${PROJECT_CP4AIOPS} | grep 'Not
Ready'
```

   Example output:

```
lifecycleservice: Not Ready
zenservice: Not Ready
```

   To see details about why a component is **Not Ready** run the following command, where **<component>** is the component that is not ready, for example **zenservice**.

```
oc get <component> -o yaml -n ${PROJECT_CP4AIOPS}
```

   (Optional) If you downloaded the status checker script earlier in step 1.1 Download documentation and scripts for offline access, then you can also run this script to see information about the status of your deployment.

If the installation fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade and Known Issues to help you identify any installation problems.

5.8 Create an EgressFirewall

There is no egress firewall policy defined when you install IBM Cloud Pak for AIOps, so outgoing traffic from workload pods to the internal and external network is unrestricted.

To create a more secure environment, use the following steps.

1. Create an **`EgressFirewall`** on your Red Hat OpenShift cluster to limit egress from the IBM Cloud Pak for AIOps project (namespace).

   For more information about creating an **`EgressFirewall`**, see Configuring an egress firewall for a project.

   Note: There must be only one **`EgressFirewall`** per project/namespace.

2. Configure exceptions to the **`EgressFirewall`**.

   Edit your **`EgressFirewall`** to add exceptions for the following IBM Cloud Pak for AIOps components that have egress dependencies, otherwise these IBM Cloud Pak for AIOps components fail when they attempt egress.

   1. Allow egress to any external services, such as the following connections:
      - Kubernetes
      - GitHub
      - Microsoft® Teams
      - ServiceNow
      - Slack
      - VMware® vCenter
   2. Configure your **`EgressFirewall`** to allow traffic for your GitHub, Kubernetes, ServiceNow, and VMware vCenter integrations.

      Edit your **`EgressFirewall`** to allow or deny egress, as in the following example:

      ```
      kind: EgressFirewall
      metadata:
        name: default
      spec:
        egress:
        - type: Allow
          to:
            cidrSelector: <1.2.3.0/24>
        - type: Allow
          to:
            dnsName: <www.github.com>
        - type: Allow
          to:
            dnsName: <www.developer.kubernetes.com>
        - type: Allow
          to:
            dnsName: <www.developer.servicenow.com>
        - type: Allow
          to:
            dnsName: <www.developer.vcenter.com>
        - type: Deny
          to:
            cidrSelector: <0.0.0.0/0>
      ```

      Substitute values for `dnsName` and `cidrSelector` that are the DNS names and addresses of your GitHub, Kubernetes, ServiceNow, or VMware vCenter sources.

5.9 Access the Automation console

After you successfully install IBM Cloud Pak for AIOps, get the URL for accessing the Automation console.

1. Use the following command to get the URL to access the Automation console:

   ```
   oc get route -n ${PROJECT_CP4AIOPS} cpd -o jsonpath='{.spec.host}'
   ```

   The following output is a sample output:

```
cpd-cp4aiops.apps.mycluster.mydomain
```

Based on the sample output, your console URL would be **https://cpd-cp4aiops.apps.mycluster.mydomain**

2. Enter the URL in your browser to open the Automation console. Log in with your username and password.

Find the IBM Cloud Pak for AIOps console username and password

The default username to access the Automation console is **admin**. You can check the default username and their password with the following commands.

Note: This information is for the **IBM provided credentials (admin only)** authentication type.

1. Find the default username.

```
oc -n ibm-common-services get secret platform-auth-idp-credentials -o
jsonpath='{.data.admin_username}' | base64 -d && echo
```

2. Get the password for the **admin** username.

```
oc -n ibm-common-services get secret platform-auth-idp-credentials -o
jsonpath='{.data.admin_password}' | base64 -d
```

The following extract shows a sample output:

```
EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA
```

Based on the sample output, your password would be **EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA**.

Important: You can change this default password at any time. For more information, see Changing the cluster administrator password.

What to do next

- Define integrations and applications with Defining.
- If you have an existing on-premises IBM Tivoli Netcool/OMNIbus deployment, then you can connect IBM Cloud Pak for AIOps to it with the Netcool connector. For more information, see Creating IBM Tivoli Netcool/OMNIbus connections.
- If you have an existing on-premises IBM Tivoli Netcool/Impact deployment, then you can connect IBM Cloud Pak for AIOps to it with the IBM Tivoli Netcool/Impact connector. For more information, see Creating IBM Tivoli Netcool/Impact connections.
- Familiarize yourself with backup and restore procedures. It is recommended that you take regular backups of your IBM Cloud Pak for AIOps deployment. For more information, see Backup and restore.

Installing on cloud platforms

When you are installing IBM Cloud Pak® for AIOps on specific cloud platforms, there are some suggested and best practices that you can follow to help you complete the installation process.

While the following suggested and required configurations are only for deployment on Azure Red Hat OpenShift (ARO), Google Cloud Platform (GCP), Red Hat® OpenShift® on IBM Cloud® service (ROKS), and Red Hat® OpenShift® Service on AWS (ROSA), IBM Cloud Pak for AIOps can be installed on any platform that has a supported version of Red Hat® OpenShift® Container Platform installed that meets all other prerequisities.

View the following topics for guidance on installing IBM Cloud Pak for AIOps on ARO, GCP, ROKS, or ROSA.

- Installing IBM Cloud Pak for AIOps on ARO
- Installing IBM Cloud Pak for AIOps on GCP
- Installing IBM Cloud Pak for AIOps on ROKS
- Installing IBM Cloud Pak for AIOps on ROSA

View the following topics for details on the main installation options for installing IBM Cloud Pak for AIOps on Red Hat OpenShift.

- Starter installation (CLI)
- Production installation

Installing IBM Cloud Pak for AIOps on Azure Red Hat OpenShift (ARO)

If you are installing IBM Cloud Pak® for AIOps on Azure Red Hat OpenShift (ARO), then you must complete some additional steps before you rejoin the main installation procedure for installing IBM Cloud Pak for AIOps on Red Hat® OpenShift® Container Platform.

Before you begin

Ensure that you meet the following prerequisites:

- Access to the Azure portal (requires email invitation).
- Red Hat subscription, with username and password.
- Azure account permissions to deploy an ARO cluster. You must have the following permissions directly on the virtual network, on the resource group, or on the subscription containing it:
  - Contributor and User Access Administrator permissions, or
  - Owner permissions

  For example, an Azure user with assigned roles of **`Application administrator`** and **`Conditional Access administrator`** has adequate permission.

Installation procedure

Follow these steps to install IBM Cloud Pak for AIOps on ARO.

1. Configure Azure and Red Hat access
2. Create an ARO cluster
3. Configure storage
4. Install IBM Cloud Pak for AIOps

1. Configure Azure and Red Hat access

1. Install the Azure command-line interface (CLI), **`az`**.

   Follow the instructions Install the Azure CLI on Linux in the Azure documentation.

2. Get an OpenShift pull secret to enable access to the OpenShift container registries.

   1. Log in to the OpenShift cluster manager portal.

   2. Click **Download pull secret** to download the pull secret, and save it to a file called **`pull-secrets.json`**. This file is used in step 2.2

3. Log in to the Azure CLI.

   Run the following command to log in to the Azure CLI, and to find your subscription-id.

   ```
   az login
   az account show | jq .id
   ```

4. Register providers to your Azure account.

   ```
   az account set --subscription <subscription-id>
   az provider register -n Microsoft.RedHatOpenShift --wait
   ```

```
az provider register -n Microsoft.Compute --wait
az provider register -n Microsoft.Storage --wait
az provider register -n Microsoft.Authorization --wait
az feature register --namespace Microsoft.RedHatOpenShift --name preview
```

Where **<subscription-id>** is the subscription-id that you found in the previous step.

2. Create an ARO cluster

1. Create dependant assets.

   Run the following commands to create a virtual network that contains two empty subnets.

```
# Set environment variables for use by the `az` commands
export SUFFIX=waiops4test
export LOCATION=eastus
export RESOURCEGROUP=aro-$SUFFIX
export CLUSTER=cluster-$SUFFIX

# Create a resource group
az group create -g $RESOURCEGROUP -l $LOCATION

# Create a new virtual network in the resource group that you created above:
az network vnet create -g $RESOURCEGROUP \
  -n aro-vnet-$SUFFIX \
  --address-prefixes 10.0.0.0/22

# Add an empty subnet for the master nodes:
az network vnet subnet create -g $RESOURCEGROUP \
  -n master-subnet-$SUFFIX \
  --vnet-name aro-vnet-$SUFFIX \
  --address-prefixes 10.0.0.0/23 \
  --service-endpoints Microsoft.ContainerRegistry \
  --disable-private-link-service-network-policies true

# Add an empty subnet for the worker nodes:
az network vnet subnet create -g $RESOURCEGROUP \
  -n worker-subnet-$SUFFIX \
  --vnet-name aro-vnet-$SUFFIX \
  --address-prefixes 10.0.2.0/23 \
  --service-endpoints Microsoft.ContainerRegistry
```

2. Run the following command to create an ARO cluster.

   Note: Use worker-vm-size **Standard_D16s_v3** for a small, starter deployment of IBM Cloud Pak for AIOps and a worker-vm-size of **Standard_D48s_v3** for a large, production deployment of IBM Cloud Pak for AIOps.

```
az aro create -g $RESOURCEGROUP \
  -n $CLUSTER \
  --vnet aro-vnet-$SUFFIX \
  --master-subnet master-subnet-$SUFFIX \
  --worker-subnet worker-subnet-$SUFFIX \
  --worker-count 6 \
  --worker-vm-size Standard_D16s_v3 \
  --master-vm-size Standard_D8s_v3 \
  --pull-secret @pull-secrets.json
```

   Note: **pull-secrets.json** must be the file that the Red Hat OpenShift secret was stored in, in step 1.2.

   It takes approximately 35 minutes for the ARO cluster to be created. Progress can be viewed from the Azure console or CLI.

3. When the ARO cluster is created, run the following command to log in to it.

```
oc login \
 -u kubeadmin -p $(az aro list-credentials  -n $CLUSTER -g $RESOURCEGROUP | jq -r
.kubeadminPassword) \
 -s $(az aro show  -n $CLUSTER -g $RESOURCEGROUP | jq -r .apiserverProfile.url)
```

3. Configure storage

Azure's native storage is not compatible with some IBM Cloud Pak for AIOps components. Configure one of the following supported RWX storage options for IBM Cloud Pak for AIOps:

- Red Hat® OpenShift® Data Foundation (ODF)
- Portworx Enterprise

3.1 Configuring Red Hat OpenShift Data Foundation

To configure Red Hat OpenShift Data Foundation storage for your IBM Cloud Pak for AIOps on ARO deployment. , use the instructions in Deploying OpenShift Data Foundation on Azure Red Hat OpenShift in the Red Hat documentation.

3.2 Configuring Portworx Enterprise

To configure Portworx storage for your IBM Cloud Pak for AIOps on ARO deployment, use the following instructions.

3.2.1 Create secret for Portworx

1. Use a browser to access the Azure portal, **`portal.azure.com`**, click **Azure Active Directory** and find the value of **Tenant ID** in the returned page.
2. From the command line, run the following command to set an environment that contains the tenant ID.

   ```
   export AZURE_TENANT_ID=<value of the tenant ID>
   ```

3. Use a browser to access the Azure portal, **`portal.azure.com`**, and then click **Virtual machines**.
4. In the search bar on the **Virtual machines** page, enter the value of your $SUFFIX environment variable as a filter and click the **Name** of any of the returned VMs.
5. Click **Access control (IAM)** on the menu.
6. On the **cluster name | Access control (IAM)** page, click **View** in the **View deny assignments** section.
7. On the **cluster name | Access control (IAM)** page's **Deny assignments** tab, click the link beneath **Name**.
8. On the **Deny assignment | Users** page, click the link in the **Name** column in the **Deny assignment excludes** section.
9. In the **enterprise application | Overview** page, locate the **Application ID** field and copy it.
10. From the command line, run the following command to set an environment variable that contains the application ID.

    ```
    export AZURE_CLIENT_ID=<value of the application ID>
    ```

11. Use a browser to navigate to the Azure portal, **`portal.azure.com`**, and then click **Azure Active Directory**.
12. On the **Azure Active Directory | Overview** page, click **App registrations** on the menu.
13. On the **Azure Active Directory | App registrations** page, enter the cluster's client-id (that you just exported) in the filter field, and then click the link in the **Display name** column.
14. On the returned **Application registration details** page, click **Certificates and secrets** on the menu.
15. In the **Certficates and secrets** page, select **+ New client secret** to create a new client secret.
16. In the **Add a client secret** page, enter a **Description** and an **Expiry**, and select **Add**.
17. On the **Certificates and secrets** page, locate the secret that you just added in the **Client secrets** section. Then select the copy icon next to the **Value** column to get the secret.
18. From the command line, run the following command to set an environment variable that contains the client secret.

    ```
    export AZURE_CLIENT_SECRET=<value of the client secret>
    ```

19. Create a secret

Run the following command to create the secret

```
oc delete secret px-azure -n kube-system
oc create secret generic px-azure \
-n kube-system \
--from-literal=AZURE_TENANT_ID=${AZURE_TENANT_ID} \
--from-literal=AZURE_CLIENT_ID=${AZURE_CLIENT_ID} \
--from-literal=AZURE_CLIENT_SECRET=${AZURE_CLIENT_SECRET}
```

3.2.2 Create a Portworx spec

A Portworx Enterprise spec is required to set up a Portworx storage cluster.

1. From a browser, go to https://portworx.com, select **LOGIN**, and log in to your Portworx account. Select to use the Portworx Enterprise edition. Then, click **Continue**, and **Continue** again on the Portworx product comparison page.
2. On the **Spec Generator - Enterprise** page, enter or select the following settings and then click **Next**.
    1. Select the checkbox for **Use the Portworx Operator**.
    2. Select *2.10* for the **Portworx Version**.
    3. Select the *Built-in* radio button for **ETCD**.
3. On the next page, set the following values and then click **Next**.
    1. For **Select your environment**, select the *Cloud* radio button.
    2. Under **Select Cloud Platform**, select *AZURE*.
    3. Under **Configure storage devices**, select *Create Using a Spec*
    4. Under **Size(GB)**, select *2000 GB*
4. Click **Next** to skip the Network configuration page.
5. On the **Customize** page, select **Azure Kubernetes Service (AKS)**, then click **Finish**, and click **Agree** to accept the license agreement.
6. In the **spec** page, highlight and select the live download URL.

    For example, **`https://install.portworx.com/2.10?operator=true&mc=false&kbver=&b=true&kd=type%3DPremium_LRS%2Csize%3D150&s=%22type%3DPremium_LRS%2Csize%3D2000%22&c=px-cluster-e0fb19ba-4e45-4175-b92e-18003948b965&aks=true&stork=true&csi=true&mon=true&tel=false&st=k8s&promop=true`**

3.2.3 Installing the Portworx operator from Red Hat OpenShift Container Platform

1. Open the Red Hat OpenShift Container Platform console for your Red Hat OpenShift Service on AWS cluster.

    If you need to find your ARO cluster credentials and console URL, you can use the following commands.

    ```
    az aro list-credentials  -n $CLUSTER -g $RESOURCEGROUP
    az aro show  -n $CLUSTER -g $RESOURCEGROUP | jq -r .consoleProfile.url
    ```

2. Go to **Operators** > **OperatorHub**.
3. Search for *Portworx Enterprise*. Then, click **Install** > **Install**.

    Warning: Do not click *Create StorageCluster*. You will create the storage cluster in the next step, with the spec that you saved.

4. *Portworx Enterprise* is now listed under **Operators** > **Installed Operators**.

3.2.4 Create the Portworx storage cluster

1. From the command line, run the following command to load the Portworx spec that you created into the ARO cluster.

    ```
    export PORTWORX_SPEC_URL=<spec_URL>
    curl -ks "${PORTWORX_SPEC_URL}&osft=true" | oc apply -f -
    ```

    Where **`<spec_URL>`** is the spec URL that you copied earlier.

2. From the console, go to **Operators** > **Installed Operators** > **Portworx Enterprise** and select the **StorageClusters** tab. The storage cluster is ready when the **Status** changes from *Phase: Initializing* to *Phase: Online*.

3. Create the Portworx storage classes by running the following commands.

   Create the `portworx-fs` storage class.

```
cat << _EOF_ | oc apply -f -
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: portworx-fs
provisioner: kubernetes.io/portworx-volume
parameters:
  repl: "3"
  io_profile: "db"
  priority_io: "high"
  sharedv4: "true"
allowVolumeExpansion: true
_EOF_
```

   Create the `portworx-aiops` storage class.

```
cat << _EOF_ | oc apply -f -
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: portworx-aiops
provisioner: kubernetes.io/portworx-volume
parameters:
  repl: "3"
  priority_io: "high"
  snap_interval: "0"
  io_profile: "db"
  block_size: "64k"
  sharedv4: "true"
allowVolumeExpansion: true
_EOF_
```

4. Install IBM Cloud Pak for AIOps

You are now ready to install IBM Cloud Pak for AIOps.

Follow the procedure for installing IBM Cloud Pak for AIOps from step 3 in one of the following topics, according to your requirements:

- Starter installation (CLI)
- Online production installation of IBM Cloud Pak for AIOps (console method)
- Online production installation of IBM Cloud Pak for AIOps (CLI method)

When you create an instance of the IBM Cloud Pak for AIOps custom resource, set the storage as follows:

For ODF storage, use

```
storageClass: ocs-storagecluster-cephfs
storageClassLargeBlock: ocs-storagecluster-ceph-rbd
```

For Portworx storage, use

```
storageClass: portworx-fs
storageClassLargeBlock: portworx-aiops
```

Deleting the ARO cluster

If you no longer require your ARO cluster, you can remove it with the following command.

```
az aro delete -y -g $RESOURCEGROUP -n $CLUSTER
```

Installing IBM Cloud Pak for AIOps on Google Cloud Platform

Follow this procedure if you want to install IBM Cloud Pak® for AIOps on a Red Hat® OpenShift® Container Platform cluster on Google Cloud Platform (GCP).

Installation procedure

1. Create a Red Hat OpenShift cluster on GCP
2. Configure storage
3. Install IBM Cloud Pak for AIOps

1. Create a Red Hat OpenShift cluster on GCP

Install a Red Hat OpenShift cluster on GCP. For more information, see the Red Hat documentation Installing on GCP.

2. Configure storage

Configure one of the following RWX storage options for IBM Cloud Pak for AIOps:

- Red Hat® OpenShift® Data Foundation (ODF)
- Portworx Enterprise

2.1 Configuring ODF

To configure ODF storage for your IBM Cloud Pak for AIOps deployment, use the instructions in the Red Hat documentation Deploying and managing OpenShift Data Foundation using Google Cloud.

Important: Read the statement about the technology preview status of Red Hat® OpenShift® Data Foundation on GCP in the previous link.

2.2 Configuring Portworx Enterprise

To configure Portworx storage for your IBM Cloud Pak for AIOps on GCP deployment, use the following instructions.

2.2.1 Create a Portworx spec

A Portworx Enterprise spec is required to set up a Portworx storage cluster. For more information, see GCP in the Portworx documentation.

1. From a browser, go to https://portworx.com, select **LOGIN**, and log in to your Portworx account. Select to use the Portworx Enterprise edition. Then, click **Continue**, and **Continue** again on the Portworx product comparison page.
2. On the **Spec Generator - Enterprise** page, enter or select the following settings and then click **Next**.
    1. Select the checkbox for **Use the Portworx Operator**.
    2. Select *2.10* for the **Portworx Version**.
    3. Select the *Built-in* radio button for **ETCD**.
3. On the next page, set the following values and then click **Next**.
    1. For **Select your environment**, select the *Cloud* radio button.
    2. Under **Select Cloud Platform**, select *Google cloud/GKE*.
    3. Under **Configure storage devices**, select *Create Using a Spec*
    4. Under **Size(GB)**, select *2000 GB*
4. Click **Next** to skip the Network configuration page.
5. On the **Customize** page, select **Google Kubernetes Engine (GKE)**, then click **Finish**, and click **Agree** to accept the license agreement.
6. In the **spec** page, highlight and select the live download URL.

For example,

```
https://install.portworx.com/2.10?
operator=true&mc=false&kbver=&b=true&kd=type%3DPremium_LRS%2Csize%3D150&s=%22type%3DPremi
um_LRS%2Csize%3D2000%22&c=px-cluster-e0fb19ba-4e45-4175-b92e-
18003948b965&aks=true&stork=true&csi=true&mon=true&tel=false&st=k8s&promop=true
```

2.2.2 Configure GCP for Portworx

1. Open the worker node ports 17001-17020 to the master node network. For more information, see Firewall policy rules in the GCP documentation.
2. The worker node's service account (created by the Red Hat OpenShift installer) must have the following roles:

   ```
   Compute Admin
   Service Account User
   Kubernetes Engine Cluster Viewer
   ```

   You can view and edit the roles by opening the GCP console, and selecting **IAM & Admin** > **Service Accounts**. For more information, see in the GCP documentation.

2.2.3 Installing the Portworx operator from Red Hat OpenShift Container Platform

1. Open the Red Hat OpenShift Container Platform console for your GCP cluster.
2. Go to **Operators** > **OperatorHub**.
3. Search for *Portworx Enterprise*. Then, click **Install** > **Install**.

   Warning: Do not click *Create StorageCluster*. You will create the storage cluster in the next step, with the spec that you saved.
4. Portworx Enterprise should now be listed under **Operators** > **Installed Operators**.

2.2.4 Create the Portworx storage cluster

1. From the command line, run the following command to load the Portworx spec that you created into the GCP cluster.

   ```
   export PORTWORX_SPEC_URL=<spec_URL>
   curl -ks "${PORTWORX_SPEC_URL}" | oc apply -f -
   ```

   Where **<spec_URL>** is the spec URL that you copied earlier.
2. From the console, go to **Operators** > **Installed Operators** > **Portworx Enterprise** and select the **StorageClusters** tab. The storage cluster is ready when the **Status** changes from *Phase: Initializing* to *Phase: Online*.
3. Create the Portworx storage classes by running the following commands.

   Create the **portworx-fs** storage class.

   ```
   cat << _EOF_ | oc apply -f -
   kind: StorageClass
   apiVersion: storage.k8s.io/v1
   metadata:
     name: portworx-fs
   provisioner: kubernetes.io/portworx-volume
   parameters:
     repl: "3"
     io_profile: "db"
     priority_io: "high"
     sharedv4: "true"
   allowVolumeExpansion: true
   _EOF_
   ```

   Create the **portworx-aiops** storage class.

```
cat << _EOF_ | oc apply -f -
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: portworx-aiops
provisioner: kubernetes.io/portworx-volume
parameters:
  repl: "3"
  priority_io: "high"
  snap_interval: "0"
  io_profile: "db"
  block_size: "64k"
  sharedv4: "true"
allowVolumeExpansion: true
_EOF_
```

3. Install IBM Cloud Pak for AIOps

You are now ready to install IBM Cloud Pak for AIOps.

Follow the procedure for installing IBM Cloud Pak for AIOps from step 3 in one of the following topics, according to your requirements:

- Starter installation (CLI)
- Online production installation of IBM Cloud Pak for AIOps (console method)
- Online production installation of IBM Cloud Pak for AIOps (CLI method)

When you create an instance of the IBM Cloud Pak for AIOps custom resource, set the storage as follows:

For ODF storage, use:

```
storageClass: ocs-storagecluster-cephfs
storageClassLargeBlock: ocs-storagecluster-ceph-rbd
```

For Portworx storage, use:

```
storageClass: portworx-fs
storageClassLargeBlock: portworx-aiops
```

Installing IBM Cloud Pak for AIOps on Red Hat OpenShift on IBM Cloud (ROKS)

If you are installing on Red Hat OpenShift on IBM Cloud (ROKS), you can follow any of the main online installation options for installing IBM Cloud Pak® for AIOps. These procedures demonstrate how to install on Red Hat OpenShift Container Platform. For installing on Red Hat OpenShift on IBM Cloud, you need to replace any storage class configuration with your configured storage class for Red Hat OpenShift on IBM Cloud.

- Starter installation (CLI)
- Production installation(online installation only)

Storage

Installing on a classic infrastructure:

When you are configuring storage as part of the installation, consider using the default storage that is available with Red Hat OpenShift on IBM Cloud (ROKS). This storage has been tested with a IBM Cloud Pak for AIOps deployment on Red Hat OpenShift on IBM Cloud. You can choose to deploy and use your own storage if needed or preferred.

The following storage classes are the default storage classes for Red Hat OpenShift on IBM Cloud:

- `storageClass: ibmc-file-gold-gid`

- `storageClassLargeBlock: ibmc-block-gold`

Note: `ibmc-file-gold-gid` does not have volume expansion enabled by default, which is required by IBM Cloud Pak for AIOps. For more information, see Storage class requirements.

Example storage configuration (classic infrastructure):

```
spec:
  automationFoundation: {}
  imagePullSecret: ibm-entitlement-key
  license:
    accept: true
  pakModules:
    - enabled: true
      name: aiManager
    - enabled: true
      name: aiopsFoundation
    - enabled: true
      name: applicationManager
    - enabled: true
      name: connection
  size: large
  storageClass: ibmc-file-gold-gid
  storageClassLargeBlock: ibmc-block-gold
status:
```

Installing on VPC infrastructure (using Red Hat OpenShift Data Foundation storage):

Consider using Red Hat OpenShift Data Foundation storage. This storage has been tested with a IBM Cloud Pak for AIOps deployment on Red Hat OpenShift on IBM Cloud. To use this storage provider, you need to first deploy Red Hat OpenShift Data Foundation on your CPV cluster. For more information, see Deploying OpenShift Data Foundation on VPC clusters.

When you do use this storage, the following storage classes are the default storage classes to use for Red Hat OpenShift on IBM Cloud:

- `storageClass: ocs-storagecluster-cephfs`
- `storageClassLargeBlock: ocs-storagecluster-ceph-rbd`

For more information about these storage classes, see the IBM Cloud Docs Storage class reference.

Example storage configuration (VPC infrastructure):

```
spec:
  automationFoundation: {}
  imagePullSecret: ibm-entitlement-key
  license:
    accept: true
  pakModules:
    - enabled: true
      name: aiManager
    - enabled: true
      name: aiopsFoundation
    - enabled: true
      name: applicationManager
    - enabled: true
      name: connection
  size: large
  storageClass: ocs-storagecluster-cephfs
  storageClassLargeBlock: ocs-storagecluster-ceph-rbd
status:
```

Installing IBM Cloud Pak for AIOps on Red Hat OpenShift Service on AWS (ROSA)

If you are installing IBM Cloud Pak® for AIOps on Red Hat OpenShift Service on AWS (ROSA), then you must complete some additional steps before you rejoin the main installation procedure for installing IBM Cloud Pak for AIOps on Red Hat® OpenShift® Container Platform.

Before you begin

Ensure that you meet the following prerequisites:

- You have an AWS account.
- You have a Red Hat subscription.
- You have a Portworx account and license. The following steps are tested with, and require, Portworx storage. If needed, register for a Portworx account.
- Ensure that your environment meets the prerequisites for Red Hat® OpenShift® Service on AWS.

Installation procedure

Follow these steps to install IBM Cloud Pak for AIOps on Red Hat OpenShift Service on AWS

1. Configure AWS and Red Hat access
2. Create a Red Hat OpenShift Service on AWS Red Hat OpenShift Container Platform cluster
3. Configure storage
4. Install IBM Cloud Pak for AIOps

1. Configure AWS and Red Hat access

1. Log in to the AWS console to verify that you have access to the console. If needed, the *Login* page provides a link to register for an account.
2. From the AWS console, deploy Red Hat OpenShift Container Platform into a VPC within your AWS environment if it is not already deployed. For more information see Red Hat OpenShift Service on AWS quickstart guide.
3. Log in to the Red Hat console to verify that you have access to the console. If needed, the *Login* page provides a link to register for an account.
4. Retrieve your Red Hat Red Hat OpenShift Cluster Manager API Token by entering the following URL:
   - https://console.redhat.com/openshift/token/rosa/show
5. Configure the ROSA CLI to work with your AWS account. For more information, see Install and configure the latest ROSA CLI.
6. (Optional) Create a test cluster to validate your environment configuration
   1. Log in to Red Hat OpenShift Service on AWS.

      ```
      rosa login --token="<token>"
      ```

   2. Configure your AWS account to allow a IAM (non-STS) ROSA cluster.

      ```
      rosa init
      ```

   3. Create the test cluster.

      ```
      rosa create cluster --cluster-name=mytest
      ```

      During the cluster creation, you can review the installation logs to watch the progress.

      ```
      rosa logs install -c mytest --watch
      ```

   4. After the installation completes, list the cluster and note the cluster ID.

      ```
      rosa list clusters
      ```

   5. Describe your cluster using the cluster ID.

```
rosa describe cluster -c 1ab23de4fghijk5lmno6p78q9r1stu2v
```

6. Add an identity provider.

   ```
   rosa create idp --cluster 1ab23de4fghijk5lmno6p78q9r1stu2v --interactive
   ```

   You need to use the noninteractive mode and get some constant values to use

7. Create your initial admin account.

   ```
   rosa create admin --cluster=1ab23de4fghijk5lmno6p78q9r1stu2v
   ```

8. Log in to your test cluster by using the admin and password.

   ```
   oc login https://api.jgtest01.dj5a.p1.openshiftapps.com:6443 --username cluster-admin --password XXXXXX
   ```

   If your login is successful, your test cluster is working and your environment is configured.

9. Clean up and delete the test cluster to proceed with creating your main cluster.

   ```
   rosa delete cluster -c 1ab23de4fghijk5lmno6p78q9r1stu2v
   ```

2. Create a Red Hat OpenShift Service on AWS Red Hat OpenShift Container Platform cluster

1. Log in to the Red Hat OpenShift Service on AWS CLI.

   ```
   rosa login
   ```

   Log in to the site https://cloud.redhat.com/openshift/token/rosa to retrieve your token. Then, copy and paste the token into the CLI prompt.

2. Create your cluster.

   ```
   rosa create cluster --cluster-name=<myclustername> --compute-machine-type=m5.8xlarge  --compute-nodes=<Number of Compute Nodes>  --version <Red Hat OpenShift_version>
   ```

   Example:

   ```
   rosa create cluster --cluster-name=cluster-test1  --compute-machine-type=m5.8xlarge  --compute-nodes=6 --version 4.10.3
   ```

3. After your cluster is ready, create your cluster administrator account.

   1. Run the following command:

      ```
      rosa create admin -c <myclustername>
      ```

      Important: Record the admin username (cluster-admin) and password for future use.

   2. Run the `oc login` command with the cluster administrator credentials.

4. Verify that all nodes are in `Ready` state before proceeding. Your cluster may take 40+ minutes to create and for you to be able to successfully log in with the 'oc login' command.

   1. Run the following command:

      ```
      oc get nodes
      ```

   2. Run the following command:

      ```
      rosa describe cluster -c <myclustername>
      ```

      Record the console URL for the Red Hat OpenShift Console and the Details Page for viewing the cluster details.

   3. Verify that you can access the Red Hat OpenShift console by logging in to the provided Red Hat OpenShift console URL using the `cluster-admin` role and credentials.

3. Configure storage

Portworx is the tested and supported storage option for installing IBM Cloud Pak for AIOps on Red Hat OpenShift Service on AWS (ROSA).

3.1 Configure the Portworx services spec

1. Log in to your Portworx account. Select to use the Portworx Enterprise edition. Then, click **Next**.
2. On the *Spec Generator - Enterprise* page, enter or select the following settings for your storage:
    1. Select the checkbox for *Use the Portworx Operator*.
    2. Select `2.10` or higher for the Portworx Version.
    3. Select the *Built-in* radio button for ETCD. Click **Next**.
    4. Select the *Cloud* radio button. Then, select *AWS* for Cloud Platform.
    5. Select the type of disk: *Create Using a Spec*
    6. Select the EBS volume type: `GP2`, Size (GB): `2000`. Click **Next**.
    7. Click **Next** to skip the Network configuration page.
    8. On the *Customize* page, select *OpenShift 4+*. Click **Finish**.
    9. Click *Agree* to accept the license agreement.
    10. Choose your own values to enter under the *Spec Name* and *Spec Tags* fields. Then, click **Save Spec**.
3. From *Spec List* page, find your Spec name and expand the Actions column menu and select **Copy to Clipboard**. Save the kubectl command. You use this command later.

3.2 Configure the AWS infrastructure for Portworx storage

1. Edit the Inbound Rules for both your master and worker nodes to allow for Network File System (NFS).
    1. Log in to the AWS EC2 Console.
    2. Under *EC2*, select *Instances*. Click an Instance ID for one of your worker nodes.
    3. Click the **Security** tab, and click the Security Group Name for the node.
    4. Click **Edit Inbound Rules**. Scroll to the bottom, and click **Add Rule**.
    5. Update the following settings for the rule:
        - Expand the first drop-down menu, and change the setting from *Custom TCP* to *All TCP*.
        - In the *CIDR blocks Source* field, enter `10.0.0.0/8`.
        - For the ports, open the following ports: 111, 2049, 20048, 17001-17020, 27017.
        - If you encounter issues during the installation, considering opening all inbound ports from `10.0.0.0/8`.
    6. Click **Save rules**.
    7. Repeat the above steps for your other nodes.
2. Create your Portworx policy in the AWS Identity and Access Management (IAM) tool.
    1. Go to **IAM** > **Access management** > **Policies**.
    2. Choose **Create policy**.
    3. Choose the **JSON** tab.
    4. Replace all the text with the following content:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "PortworxPolicy",
      "Effect": "Allow",
      "Action": [
        "ec2:AttachVolume",
        "ec2:ModifyVolume",
        "ec2:DetachVolume",
        "ec2:CreateTags",
        "ec2:CreateVolume",
        "ec2:DeleteTags",
        "ec2:DeleteVolume",
        "ec2:DescribeTags",
        "ec2:DescribeVolumeAttribute",
```

```
        "ec2:DescribeVolumesModifications",
        "ec2:DescribeVolumeStatus",
        "ec2:DescribeVolumes",
        "ec2:DescribeInstances",
        "autoscaling:DescribeAutoScalingGroups"
      ],
    "Resource": [
      "*"
    ]
  }
]
}
```

5. Click **Next: Tags**. Then, click **Next: Review**.

6. Enter the name for your new policy. Then, click **Create policy**.

3. Attach the policy.

    1. Go to **IAM** > **Roles**.
    2. Click the worker-role name for your cluster.
    3. Click **Attach policy**. Search for, and select, the policy that you created. Then, select to attach the policy.

3.3 Install the Portworx operator from Red Hat OpenShift Container Platform

1. Open the Red Hat OpenShift Container Platform console for your Red Hat OpenShift Service on AWS cluster.
2. Go to **Operators** > **OperatorHub**.
3. Search for Portworx Enterprise. Then, click **Install** > **Install**.
4. Portworx Enterprise is now be listed under *Installed Operators*.

3.4. Configure the Portworx operator

1. Import the Spec from Portworx.

    1. Run `oc login` to log in to your Red Hat OpenShift Service on AWS cluster.
    2. Run `kubectl apply -f <command>` where `<command>` is the command that you saved earlier from the Portworx console.
2. Verify that Portworx Enterprise shows as one of the Installed Operators in `Project: kube-system`. Click Portworx Enterprise tile.

    In the **Storage Cluster** tab, you can see the storage cluster being initialized. Wait until Status is `Phase: Online`.

3. Create the Portworx storage classes by running the following commands.

    Create the `portworx-fs` storage class.

```
cat << _EOF_ | oc apply -f -
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: portworx-fs
provisioner: kubernetes.io/portworx-volume
parameters:
  repl: "3"
  io_profile: "db"
  priority_io: "high"
  sharedv4: "true"
allowVolumeExpansion: true
_EOF_
```

    Create the `portworx-aiops` storage class.

```
cat << _EOF_ | oc apply -f -
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: portworx-aiops
provisioner: kubernetes.io/portworx-volume
parameters:
```

```
  repl: "3"
  priority_io: "high"
  snap_interval: "0"
  io_profile: "db"
  block_size: "64k"
  sharedv4: "true"
allowVolumeExpansion: true
_EOF_
```

4. Install IBM Cloud Pak for AIOps

You are now ready to install IBM Cloud Pak for AIOps.

Follow the procedure for installing IBM Cloud Pak for AIOps from step 3 in one of the following topics, according to your requirements:

- Starter installation (CLI)
- Online production installation of IBM Cloud Pak for AIOps (console method)
- Online production installation of IBM Cloud Pak for AIOps (CLI method)

When you create an instance of the IBM Cloud Pak for AIOps custom resource, set the storage as follows:

```
storageClass: portworx-fs
storageClassLargeBlock: portworx-aiops
```

Deleting the Red Hat OpenShift Service on AWS cluster

If you no longer require your Red Hat OpenShift Service on AWS cluster, you can remove it with the following steps.

1. Run `oc login` to log in to your Red Hat OpenShift Service on AWS cluster.
2. Run the following command to delete the cluster:

   ```
   rosa delete cluster -c <clustername> --watch
   ```

3. Run the following command to verify that the cluster is deleted:

   ```
   rosa list clusters
   ```

   Ensure that your cluster no longer is listed.

4. Delete any Identity and Access Management (IAM) policies for the cluster.

Installing Infrastructure Automation

Follow these steps to install Infrastructure automation.

If you have a license for IBM Cloud Pak® for AIOps, you are entitled to install and use Infrastructure Automation. Choose one of the following deployment options:

- Online installation of Infrastructure Automation for use with IBM Cloud Pak for AIOps
- Offline installation of Infrastructure Automation for use with IBM Cloud Pak for AIOps
- Online installation of Infrastructure Automation
- Offline installation of Infrastructure Automation
  - Bastion Host
  - Portable Compute Device
  - Portable storage device

Installing stand-alone Infrastructure Automation

Follow these instructions for installing a stand-alone instance of Infrastructure Automation.

- Offline installation
- Online installation
- Post installation
  - Installation and Upgrade
  - Installing on AWS
  - Installing on VMware vSphere
  - Installing on Red Hat OpenStack Platform
  - Integration with OpenShift Container Platform
  - High Availability Guide
  - Upgrading to Infrastructure Management appliance
  - Migrating from an Infrastructure management appliance to a Kubernetes (podified) installation
  - Appliance Hardening Guide

Online installation of Infrastructure Automation

Follow these steps to install Infrastructure automation.

Infrastructure automation is included with IBM Cloud Pak® for AIOps. If you have a license for IBM Cloud Pak for AIOps, you are entitled to install and use Infrastructure automation.

Before you begin

Confirm that your environment meets the requirements for Infrastructure Automation and your chosen installation method:

- Review the Planning section.
- You need to make sure that your system has the minimum requirements needed for installing Infrastructure Automation. For instance, your architecture must be AMD64 or Linux® on Power® (ppc64le) or Linux® on IBM Z® and LinuxONE.

Note: The display names of some OpenShift console components, such as window titles and push buttons, vary between OpenShift versions. The following instructions are based on OpenShift version 4.8 console components.

Procedure

From a high level, an installation of Infrastructure Automation consists of the following steps:

1. Install and configure Red Hat OpenShift
2. Configure storage
3. Create a custom project (namespace)
4. Create the catalog source
5. Create the entitlement key secret
6. Create an EgressFirewall
7. Install the Infrastructure Automation operator
8. Install Infrastructure Automation
9. Verify the install
10. Check the deployed pods in the deployed namespace
11. Log in to the Infrastructure Automation console
12. Assign user roles and permissions

1. Install and configure Red Hat OpenShift Container Platform

For more information about the supported OpenShift versions, see Supported Red Hat OpenShift Container Platform versions.

1. Install OpenShift by using the instructions in Installing Red Hat OpenShift Container Platform.

2. Install the OpenShift (`oc`) command-line interface (CLI) on your cluster's boot node by using the instructions in Getting started with the OpenShift CLI, and then run `oc login`.

3. Optionally configure a custom certificate for Infrastructure Automation to use. You can use either of the following methods:

    - Configure a custom certificate for the Red Hat OpenShift cluster. Follow the instructions in the Red Hat OpenShift documentation Replacing the default ingress certificate. Then, deploy the signing CA certificate into the cluster by following the instructions in the Red Hat OpenShift documentation Replacing the CA Bundle certificate.
    - If you would like to use a custom certificate for Infrastructure Automation only, then after installation is complete follow the instructions in Using a custom certificate.

2. Configure storage

You must configure your own storage for use with Infrastructure Automation. For more information, see Storage considerations.

3. Create a custom project (namespace)

Create a project (namespace) called `cp4aiops` for your Infrastructure Automation deployment, by running the following command:

```
oc create namespace cp4aiops
```

Infrastructure Automation does not support deploying into different namespaces within a cluster. It has to be the same namespace.

4. Create the catalog source

You must add the Infrastructure Automation catalog source to your OpenShift cluster.

1. Log in to your OpenShift cluster's console.

2. Add the `IBM Operators` CatalogSource.

    The `ibm-operator-catalog` CatalogSource object can be configured to automatically poll for a newer version, and to retrieve it if one is available. This triggers an automatic update of your Infrastructure Automation deployment. Polling for updates is enabled by configuring the polling attribute, `spec.updateStrategy.registryPoll`.

    Note: `ibm-operator-catalog` also contains the catalogs for other CloudPaks. If you have multiple CloudPaks installed on your cluster then an automatic update will be configured for all of them.

    Click the plus icon in the upper right corner to open the **Import YAML** dialog box, paste in one of the following codeblocks, and then click **Create**.

    If you do not want to enable automatic updates, use this YAML:

    ```
    apiVersion: operators.coreos.com/v1alpha1
    kind: CatalogSource
    metadata:
      name: ibm-operator-catalog
      namespace: openshift-marketplace
    spec:
      displayName: ibm-operator-catalog
      publisher: IBM Content
      sourceType: grpc
      image: icr.io/cpopen/ibm-operator-catalog:latest
    ```

    If you want to enable automatic updates, use this YAML:

    ```
    apiVersion: operators.coreos.com/v1alpha1
    kind: CatalogSource
    metadata:
      name: ibm-operator-catalog
      namespace: openshift-marketplace
    spec:
      displayName: ibm-operator-catalog
      publisher: IBM Content
    ```

```
  sourceType: grpc
  image: icr.io/cpopen/ibm-operator-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
```

3. Go to **Administration** > **Cluster Settings**. Under **Global Configuration** > **OperatorHub** > **Sources**, verify that the *ibm-operator-catalog* CatalogSource object is present.

5. Create the entitlement key pull secret

1. Log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.
2. In the *Entitlement keys* section, select **Copy key** to copy your entitlement key to the clipboard.
3. From the OpenShift CLI, run the following command:

```
oc create secret docker-registry ibm-entitlement-key \
    --docker-username=cp\
    --docker-password=<entitlement-key> \
    --docker-server=cp.icr.io \
    --namespace=cp4aiops
```

Where **<entitlement-key>** is the entitlement key that you copied in the previous step.

6. Create an EgressFirewall

There is no egress firewall policy defined when you install Infrastructure Automation, so outgoing traffic from workload pods to the internal and external network is unrestricted.

If you require a more secure environment, then use the following steps.

1. Create an **EgressFirewall** on your Red Hat OpenShift cluster to limit egress from the Infrastructure Automation project (namespace).

   For information on creating an **EgressFirewall**, see Configuring an egress firewall for a project.

   Note: There must be only one **EgressFirewall** per project (namespace).

2. Configure exceptions to the **EgressFirewall**.

   You must edit your **EgressFirewall** to allow traffic for external services, outbound cloud connections, and Infrastructure Management providers that you manage that have egress dependencies, otherwise these components fail when attempting egress.

   For more information about Infrastructure Automation connections and providers, see Managing connections and Managing Providers.

   Edit your **EgressFirewall** to allow or deny egress, as in the following example:

```
kind: EgressFirewall
metadata:
  name: default
spec:
  egress:
  - type: Allow
    to:
      cidrSelector: <1.2.3.0/24>
  - type: Allow
    to:
      dnsName: <www.example.com>
  - type: Allow
    to:
      dnsName: <www.developer.kubernetes.com>
  - type: Deny
    to:
      cidrSelector: <0.0.0.0/0>
```

Where the values you enter for **`dnsName`** and **`cidrSelector`** are the DNS names and addresses of sources that you require access for.

7. Install the Infrastructure Automation operator

Note: Both the operators **`iaf-core-operators`** and **`iaf-operators`** need to be pre-installed to prevent Cloud Pak for AIOps from crashing. Install the Infrastructure Automation operator for a Power cluster.

For more information about operators, see Adding Operators to a cluster in the Red Hat OpenShift documentation.

1. Log in to your OpenShift cluster's console.
2. Click **Operators > OperatorHub.** The **OperatorHub** page is displayed.
3. In the **All Items** field, enter **`IBM Infrastructure Automation.`** The Infrastructure Automation operator is displayed.
4. Click the **IBM Infrastructure Automation** tile. The **IBM Infrastructure Automation** window is displayed.
5. Click **Install**. The **Install Operator page** is displayed.
6. Enter the following values:
    - Set the **Namespace** to be the project (namespace) in which to install the Operator, such as cp4aiops.
    - Set **Update Channel** to v4.2
    - Set **Approval Strategy** to *Automatic.*
7. Click **Install** and wait for the IBM Infrastructure Automation operator to install.
8. Verify that the IBM Infrastructure Automation is successfully installed.

Navigate to **Operators > Installed Operators**, and select your project from the **Projects** dropdown. IBM Infrastructure Automation and its dependent operators in the project are listed with a status of *Succeeded.*

After a few minutes, verify that the Infrastructure Automation operator is installed in the **`cp4aiops`** project (namespace) with the following command:

```
oc get pods -n cp4aiops | grep ibm-infrastructure-automation-operator-controller-manager
```

8. Install Infrastructure Automation

You can either create Infrastructure Automation custom resource with the default set of values that are provided by default or customize these value before you create Infrastructure Automation custom resource.

You would typically change the default value of Infrastructure Automation custom resource, if you plan to change one or more of the following for Managed services during installation of Managed services:

- Default pre-created secrets in Managed services
- DefaultManaged services proxy setting
- Default Managed services default github.com token to import templates
- Default HTTPS certificate for Managed services Management Console
- Configuring Managed services to set the SSH keys and known hosts in provider terraform pod
- Modify the default replica count of the pods in Managed services

Create Infrastructure Automation custom resource with customization

The following command creates an instance of the Infrastructure Automation custom resource called **`IAConfig`** with customization to Managed services. You must have your customization parameters under **`spec.manageservice`** section. The list of install parameters that can be customized along with its default values are listed in Managed services installation parameters page.

```
cat << EOF | oc apply -f -
kind: IAConfig
apiVersion: aiops.ibm.com/v1alpha1
metadata:
  name: ibm-ia-installer
  namespace: cp4aiops
spec:
```

```
  imagePullSecret: ibm-entitlement-key
  infraAutoComposableComponents:
  - enabled: <Set to true to install Infrastructure Management component of Infrastructure
Automation, false otherwise>
    name: ibm-management-im-install
    spec: {}
  - enabled: <Set to true to install Managed services component of Infrastructure Automation,
false otherwise>
    name: ibm-management-cam-install
    spec:
      manageservice:
        <Set your custom installation parameter values>
  license:
  accept: <Set true to accept the license>
  storageClass: <Storage Class name that supports RWX>
  storageClassLargeBlock: <Select a storage class with a large block size (for example, 64k)>
EOF
```

Create Infrastructure Automation custom resource with default values

Run the following command to create an instance of the Infrastructure Automation custom resource called **IAConfig**. The list of install parameters with its default values are listed in Managed services installation parameters page.

```
cat << EOF | oc apply -f -
kind: IAConfig
apiVersion: aiops.ibm.com/v1alpha1
metadata:
  name: ibm-ia-installer
  namespace: cp4aiops
spec:
  imagePullSecret: ibm-entitlement-key
  infraAutoComposableComponents:
    - enabled: <set to true to install Infrastructure Management component of Infrastructure
Automation, false otherwise>
      name: ibm-management-im-install
      spec: {}
    - enabled: <set to true to install Managed services component of Infrastructure
Automation, false otherwise>
      name: ibm-management-cam-install
      spec: {}
  license:
    accept: <set true to accept the license>
  storageClass: <Storage Class name that supports RWX>
  storageClassLargeBlock: <Select a storage class with a large block size (for example, 64k)>
EOF
```

Create Infrastructure Automation custom resource and deploying Infrastructure management operand.

Run the following command to create an instance of the Infrastructure Automation custom resource called IAConfig. The list of install parameters will deploy Infrastructure management operand.

```
cat << EOF | oc apply -f -
kind: IAConfig
apiVersion: aiops.ibm.com/v1alpha1
metadata:
  name: ibm-ia-installer
  namespace: cp4aiops
spec:
  imagePullSecret: ibm-entitlement-key
  infraAutoComposableComponents:
    - enabled: <set to true to install Infrastructure Management component of Infrastructure
Automation, false otherwise>
      name: ibm-management-im-install
      spec:
        iminstall:
          applicationDomain: <YOUR_IM_HTTPD_ROUTE>
          imagePullPolicy: Always
          imagePullSecret: ibm-entitlement-key
          initialAdminGroupName: <YOUR_LDAP_USER_GROUP>
          license:
```

```
            accept: true
    - enabled: <set to true to install Managed services component of Infrastructure
Automation, false otherwise>
      name: ibm-management-cam-install
      spec: {}
  license:
    accept: <set true to accept the license>
  storageClass: <Storage Class name that supports RWX>
  storageClassLargeBlock: <Select a storage class with a large block size (for example, 64k)>
EOF
```

Where:

- YOUR_IM_HTTPD_ROUTE is a user-defined route, which must include a name for your installation plus part of your Infrastructure Automation console route. Use the following oc command to obtain the console route and modify it to derive your Infrastructure Automation route.

  ```
  oc get ingress.config.openshift.io -o=jsonpath='{.items[0].spec.domain}'
  ```

  Example output:

  ```
  apps.mycluster.myibm.com
  ```

- Add inframgmtinstall to the output to create **YOUR_IM_HTTPD_ROUTE**

  For example: **inframgmtinstall.apps.mycluster.myibm.com**

- YOUR_LDAP_USER_GROUP is an existing user-group defined in your LDAP repository. As part of the initial setup, this LDAP group is created in Infrastructure Automation to match your existing LDAP group by name, and assigned an account role which facilitates SSO login.

Important:

- You must specify an LDAP user-group and it must contain at least one user that is able to login to the Infrastructure Automation console. For example, you have an existing LDAP group that is named group100 and a user with the username user100 is a member of the group. You enter group100 for the value of <YOUR_LDAP_USER_GROUP>.

- YOUR_IM_HTTPD_ROUTE is a user-defined route, which must include a name for your installation plus part of your Infrastructure Automation console route. Use the following oc command to obtain the console route and modify it to derive your Infrastructure Automation route.

  ```
  oc -n ibm-common-services get route cp-console --template `{{.spec.host}}`
  ```

  Example output:

  ```
  cp-console.apps.mycluster.myibm.com
  ```

  Replace the first section cp-console with inframgmtinstall to create YOUR_IM_HTTPD_ROUTE

  For example: **inframgmtinstall.apps.mycluster.myibm.com**

- **YOUR_LDAP_USER_GROUP** is an existing user-group defined in your LDAP repository. As part of the initial setup, this LDAP group is created in Infrastructure Automation to match your existing LDAP group by name, and assigned an account role which facilitates SSO login.

Important: After deploying the Infrastructure management operand, make sure the LDAP group is added to the Platform UI by navigating to **Administration > Access control**, and then create a new LDAP group from existing LDAP group. Add the LDAP group to the Platform UI. For more information, see Configuring LDAP connections.

9. Verify the install

After a few minutes, run the following command to verify that your deployment is successful.

```
oc get iaconfigs.aiops.ibm.com -A
```

Example output:

```
NAME                PHASE         VERSION   STORAGECLASS   STORAGECLASSLARGEBLOCK   AGE
ibm-ia-installer    Installing    4.2.0     rook-cephfs    rook-ceph-block          8m44s
```

Note: The STORAGECLASSLARGEBLOCK will be blank if the environment is upgraded from a previous release.

There is no egress firewall policy defined when you install Infrastructure Automation, so outgoing traffic from workload pods to the internal and external network is unrestricted. There is a need to create an Egress network policy to make your environment more secure. Follow the steps here to create an Egress network policy.

10. Check the deployed pods in the deployed namespace

```
oc -n cp4aiops get deploy | grep -E "cam|ibm-infra"
```

Example output:

```
cp4aiops     cam-iaas                                  1/1     1
1           12h
cp4aiops     cam-install-operator-controller-manager   1/1     1
1           12h
cp4aiops     cam-mongo                                 1/1     1
1           12h
cp4aiops     cam-orchestration                         1/1     1
1           12h
cp4aiops     cam-portal-ui                             1/1     1
1           12h
cp4aiops     cam-provider-ansible                      0/0     0
0           12h
cp4aiops     cam-provider-bpm                          0/0     0
0           12h
cp4aiops     cam-provider-terraform-api                1/1     1
1           12h
cp4aiops     cam-provider-terraform-runtime            0/0     0
0           12h
cp4aiops     cam-proxy                                 1/1     1
1           12h
cp4aiops     cam-service-composer-api                  1/1     1
1           12h
cp4aiops     cam-service-composer-ui                   1/1     1
1           12h
cp4aiops     cam-service-library-ui                    1/1     1
1           12h
cp4aiops     cam-service-library-ui-api                1/1     1
1           12h
cp4aiops     cam-tenant-api                            1/1     1
1           12h
cp4aiops     cam-ui-basic                              1/1     1
1           12h
cp4aiops     ibm-infra-management-install-operator     1/1     1
1           12h
```

11. Log in to the Infrastructure Automation console

1. Find the password for the *admin* username by running the following command:

   ```
   oc -n ibm-common-services get secret platform-auth-idp-credentials -o
   jsonpath='{.data.admin_password}' | base64 -d
   ```

2. Find the URL to access the Infrastructure Automation console with the following command.

   ```
   oc get route -n cp4aiops cpd -o jsonpath={.spec.host}
   ```

   The following output is a sample output:

   ```
   cpd-cp4aiops.apps.mycluster.mydomain
   ```

   Based on the sample output, your console URL would be **`https://cpd-cp4aiops.apps.mycluster.mydomain`**

12. Assign user roles and permissions

When you install Infrastructure Automation and deploy Infrastructure management, you, or an administrator, must add the required Kubernetes permissions to user roles before your users can begin to access and use Infrastructure Automation tools, such as Managed services or the Service catalog. For instance, users that do not have an Administrator role are not able to use

the Infrastructure management Managed services and Service Catalog or create user groups. For more information about how to add permissions to a role, see Managing roles for Infrastructure Automation.

Installing IBM Cloud Pak for AIOps Infrastructure Automation in an air-gapped environment (offline)

If your cluster is not connected to the internet, you can complete a production install of IBM Cloud Pak® for AIOps Infrastructure Automation in your cluster by using either a bastion host, portable compute device, or a portable storage device.

Overview

It is common in production to have a cluster that does not have internet access. In these cases, you can still install Infrastructure Automation and Red Hat® OpenShift® Container Platform in an air-gapped (otherwise known as offline or disconnected) environment. Unlike online installations, air-gapped installations require you to enable the IBM Operator Catalog to mimic a typical online installation by using images in your own registry.

The following diagram provides an overview of air-gapped installation scenarios for portable compute, portable storage, and bastion hosts:

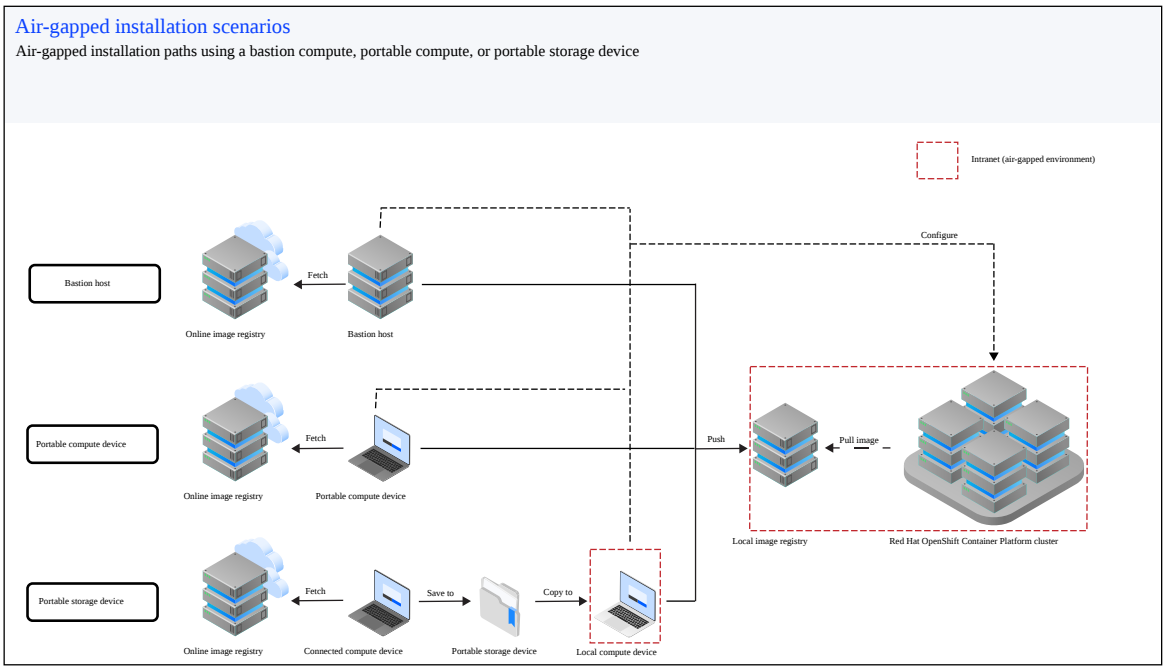


Figure. Air-gap scenario

All of these scenarios use Container Application Software for Enterprises (CASE) files to mirror content from a source to a target. CASE is a specification that defines metadata and structure for packaging, managing, and unpacking containerized applications.

You can store the product code and images to a portable compute device (like a laptop), portable storage device (like an external hard disk drive), or a bastion host and then transfer them to a local air-gapped network. If you are using either a portable compute device or a portable storage device, you don't need a bastion host to install in your air-gapped environment.

The images that are stored are large, and you must ensure that each of your airgap devices have enough storage, as follows.

- Bastion host method - the local registry and the bastion host must each have at least 58 GB of storage space.
- Portable compute device method - the local registry and the portable compute device must each have at least 58 GB of storage space.
- Portable storage device - the local registry, connected compute device, portable storage device, and the local compute device must each have at least 58 GB of storage space.

Table 1. Air-gapped installation scenarios

Air-gapped installation approach	Example	Description
Bastion host	Bastion host	A bastion server is a device that has access to both the public internet and the local intranet where a local registry and Red Hat OpenShift Container Platform clusters are located. Using the bastion server, you can replicate your images through the bastion server directly to the local, intranet registry behind the firewall.
Portable compute device	Laptop	A portable compute device, such as a laptop, can be used to download images from the entitled registry to a portable container registry running locally on the device. You can then bring the device behind your firewall and copy the images from your portable registry on the device to the local, intranet registry behind the firewall.
Portable storage device	Portable hard disk drive	Images are mirrored to a container registry on a connected compute device, and then copied to a portable storage device. The portable storage device is plugged into a local compute device inside your firewall, and the images are copied to the local compute device. The images from the local compute device are then mirrored into the local, intranet registry behind the firewall.

From a high level, an air-gapped installation of Infrastructure Automation consists of five steps:

1. Set up your image registry access and mirroring environment (One-time action)
2. Set environment variables and download CASE files
3. Mirror images depending on installation scenario
4. Configure storage
5. Install Infrastructure Automation by way of Red Hat OpenShift Container Platform

Installation methods

The following sections illustrate the required steps for each approach to an air-gapped installation. Pick the appropriate methodology and follow the steps within that section. Step ordering might vary depending on your methodology:

- Bastion host
- Portable compute device
- Portable storage device

Installing and upgrading Infrastructure management components

Instructions related to installing and upgrading to Infrastructure Automation - Infrastructure management.

Install the Infrastructure management appliance

Installing the Infrastructure management appliance consists of these steps:

1. Downloading the Infrastructure management appliance package for your environment. Download the Infrastructure management appliance
2. Setting up a virtual machine based on the appliance. For more information, see the installation topic for your platform.
3. Configuring the Infrastructure management appliance. Follow the steps in the installation topic for your platform.
    - Installing Infrastructure management on Amazon Elastic Compute Cloud (Amazon EC2)
    - Installing Infrastructure management on VMware vSphere
    - Installing Infrastructure management on Red Hat OpenStack Platform
    - Installing Infrastructure management on Red Hat Virtualization

Downloading Infrastructure management packages

Complete the following procedure to download your packages:

1. Go to the IBM® Passport Advantage® Online tab at Passport Advantage and log in with your IBM ID.
2. Find your part number.
3. Search for the files by using the part numbers that are listed in the table.
4. Download the files to a directory on your system.

For more information about the available packages for Infrastructure Automation, including for Infrastructure management, see Passport Advantage part numbers.

- Installing on Amazon Elastic Compute Cloud (Amazon EC2)
  - Installing Infrastructure Automation
    - Obtaining the appliance
    - Requirements
    - Infrastructure Automation Requirements
      - Amazon EC2 Requirements
    - Uploading the Appliance to an Amazon S3 Bucket
    - Configuring Amazon EC2 to Import the Appliance
    - Importing the Appliance to Amazon Elastic Compute Cloud (Amazon EC2)
  - Configuring Infrastructure Automation
    - Accessing the Appliance Console
    - Configuring a Database
      - Configuring an Internal Database
      - Configuring an External Database
    - Configure Messaging
    - Configuring a Worker Appliance
- Appendix
- Appliance Console Command-Line Interface (CLI)
  - Database Configuration Options
  - v2_key Options
  - IPA Server Options
  - Certificate Options
  - Other Options

Installing on Amazon Elastic Compute Cloud (Amazon EC2)

Installing Infrastructure Automation

Installing Infrastructure Automation consists of the following steps:

1. Downloading the appliance for your environment as a virtual machine snapshot template.
2. Setting up a virtual machine based on the appliance.
3. Configuring the Infrastructure Automation appliance.

After you have completed all the procedures in this guide, you will have a working environment on which additional customizations and configurations can be performed.

Obtaining the appliance

Requirements

Below are the two sets of requirements for installing Infrastructure Automation on Amazon EC2.

Infrastructure Automation Requirements

- 44 GB of space on the chosen datastore.
- 12 GB RAM.
- 4 vCPUs.
- Check that there are inbound rules to access ports 443, 22, and 80.

Amazon EC2 Requirements

- An Amazon S3 bucket to store the disk image that will be imported to AWS as a snapshot.
- A VM import service role (IAM role) named `vmimport`.

For information on creating an Amazon S3 bucket and a VM Import Service Role, see the Amazon EC2 documentation.

Uploading the Appliance to an Amazon S3 Bucket

From your local file system, you can now upload the Infrastructure Automation appliance VHD image obtained in Obtaining the appliance to the Amazon S3 bucket, using your choice of tool, such as AWS client or directly upload the Appliance image from your S3 bucket on AWS Management Console.

Configuring Amazon EC2 to Import the Appliance

Important:

These are the procedural steps as of the time of writing. For the latest information on importing a virtual machine as an image, see the Amazon EC2 documentation.

1. Install the AWS client on the computer you want to interact with the AWS API from.

   ```
   $ pip install awscli
   ```

2. Configure and download your AWS secret/access key by following the steps in the Managing Access Keys for Your AWS Account documentation.

3. Configure the AWS client with your access/secret key. For example:

   ```
   $ aws configure
   AWS Access Key ID [******]: ACCESS_KEY
   AWS Secret Access Key [******]: SECRET_KEY
   Default region name [None]:
   Default output format [None]:
   ```

4. Create the `trust-policy.json` file for the **vmimport** role. For example:

   ```
   {
      "Version": "2012-10-17",
      "Statement": [
         {
            "Effect": "Allow",
            "Principal": { "Service": "vmie.amazonaws.com" },
            "Action": "sts:AssumeRole",
            "Condition": {
               "StringEquals":{
                  "sts:Externalid": "vmimport"
               }
            }
         }
      ]
   }
   ```

5. Create the **vmimport** role using the `trust-policy.json` file that you just created.

   ```
   $ aws iam create-role --role-name vmimport --assume-role-policy-document file://trust-policy.json
   ```

Note:

This user must have permissions to create and modify IAM roles.

6. Create the **role-policy.json** file. Be sure to use the exact S3 bucket name. For example:

```
{
  "Version": "2012-10-17",
 "Statement": [
   {
     "Effect": "Allow",
     "Action": [
       "s3:ListAllMyBuckets"
     ],
     "Resource": "*"
   },
   {
     "Effect": "Allow",
     "Action": [
       "s3:CreateBucket",
       "s3:DeleteBucket",
       "s3:DeleteObject",
       "s3:GetBucketLocation",
       "s3:GetObject",
       "s3:ListBucket",
       "s3:PutObject",
       "s3:GetBucketAcl"
     ],
     "Resource":
["arn:aws:s3:::BUCKET_TO_UPLOAD_IMAGE","arn:aws:s3:::BUCKET_TO_UPLOAD_IMAGE/*"]
   },
   {
     "Effect": "Allow",
     "Action": [
       "iam:CreateRole",
       "iam:PutRolePolicy"
     ],
     "Resource": "*"
   },
   {
     "Effect": "Allow",
     "Action": [
       "ec2:ModifySnapshotAttribute",
       "ec2:CopySnapshot",
       "ec2:RegisterImage",
       "ec2:Describe*"
     ],
     "Resource": "*"
   },
   {
     "Effect": "Allow",
     "Action": [
       "ec2:CancelConversionTask",
       "ec2:CancelExportTask",
       "ec2:CreateImage",
       "ec2:CreateInstanceExportTask",
       "ec2:CreateTags",
       "ec2:DeleteTags",
       "ec2:DescribeConversionTasks",
       "ec2:DescribeExportTasks",
       "ec2:DescribeExportImageTasks",
       "ec2:DescribeInstanceAttribute",
       "ec2:DescribeInstanceStatus",
       "ec2:DescribeInstances",
       "ec2:DescribeTags",
       "ec2:ExportImage",
       "ec2:ImportInstance",
       "ec2:ImportVolume",
       "ec2:StartInstances",
       "ec2:StopInstances",
       "ec2:TerminateInstances",
       "ec2:ImportImage",
```

```
        "ec2:ImportSnapshot",
        "ec2:DescribeImportImageTasks",
        "ec2:DescribeImportSnapshotTasks",
        "ec2:CancelImportTask",
        "ec2:DescribeImages",
        "ec2:DescribeSnapshots"
    ],
    "Resource": "*"
    }
  ]
}
```

7. Apply the **vmimport** role to the Infrastructure Automation appliance image you uploaded to the S3 bucket.

```
 $ aws iam put-role-policy --role-name vmimport --policy-name vmimport --policy-document
file://role-policy.json
```

Importing the Appliance to Amazon Elastic Compute Cloud (Amazon EC2)

To import the appliance:

1. Create a containers.json file:

```
{
    "Description": "NAME OF IMPORTED SNAPSHOT IN AWS",
    "Format": "vhd",
    "UserBucket": {
        "S3Bucket": "BUCKET WITH UPLOADED .VHD IMAGE",
        "S3Key": "PATH OF .VHD IMAGE"
    }
}
```

   See the AWS documentation on VM import and export requirements, such as image formats, instances, volume and file system types, and using regions.

2. Use the AWS-CLI tools to import a disk as a snapshot. See the AWS documentation on using VM Import/Export to import a disk as a snapshot.

   Note:

   You can either specify a region, or if not, ensure that the S3 bucket is in the same region where you want to import the snapshot.

```
 $ aws ec2 import-snapshot --disk-container file://containers.json
```

3. Check the progress of your snapshot import by running the following command:

```
 $ aws ec2 describe-import-snapshot-tasks --import-task-ids SNAPSHOT_ID_GOT_FROM_RESPONSE
```

4. Create an AMI from the snapshot. See the AWS documentation on using options with the following command to create and register a Linux AMI from a snapshot.

```
 $ aws ec2 register-image
```

Configuring Infrastructure Automation

After installing Infrastructure Automation and running it for the first time, you must perform some basic configuration. You must complete these steps:

1. Add a disk to the infrastructure that is hosting your appliance.
2. Configure the database.
3. Configure messaging

Configure the appliance by using the internal appliance console.

Accessing the Appliance Console

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The Infrastructure Automation appliance summary screen displays.
3. Press **`Enter`** to manually configure settings.
4. Press the number for the item you want to change, and press **`Enter`**. The options for your selection are displayed.
5. Follow the prompts to make the changes.
6. Press **`Enter`** to accept a setting where applicable.

Note:

The Infrastructure Automation appliance console automatically logs out after five minutes of inactivity.

Configuring a Database

Infrastructure Automation uses a database to store information about the environment. Before using Infrastructure Automation, configure the database options for it; Infrastructure Automation provides the following two options for database configuration:

Configuring an Internal Database

Before installing an internal database, add a disk to the infrastructure hosting your appliance. See the documentation specific to your infrastructure for instructions for adding a disk. As a storage disk usually cannot be added while a virtual machine is running, Red Hat recommends adding the disk before starting the appliance. Infrastructure Automation only supports installing of an internal VMDB on blank disks; installation will fail if the disks are not blank.

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The Infrastructure Automation appliance summary screen displays.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. You are prompted to create or fetch an encryption key.
    - If this is the first Infrastructure Automation appliance, choose **Create key**.
    - If this is not the first Infrastructure Automation appliance, choose **Fetch key from remote machine** to fetch the key from the first appliance. For worker and multi-region setups, use this option to copy key from another appliance.

      Note:

      All Infrastructure Automation appliances in a multi-region deployment must use the same key.
6. Choose **Create Internal Database** for the database location.

   In the **Configure Messaging** menu, select **Make No messaging changes**. If you see **`Configuration failed: Internal database require a volume mounted at /var/lib/pgsql. Please add an unpartitioned disk and try again.`** message, then ensure to add a second disk for the database per instructions as defined above.
7. Choose a disk for the database. This can be either a disk you attached previously, or a partition on the current disk.

   Red Hat recommends using a separate disk for the database.
   If there is an unpartitioned disk attached to the virtual machine, the dialog will show options similar to the following:

```
1) /dev/vdb: 20480
2) Don't partition the disk
```

- Enter **1** to choose **/dev/vdb** for the database location. This option creates a logical volume using this device and mounts the volume to the appliance in a location appropriate for storing the database. The default location is **/var/lib/pgsql**, which can be found in the environment variable **$APPLIANCE_PG_MOUNT_POINT**.
- Enter **2** to continue without partitioning the disk. A second prompt will confirm this choice. Selecting this option results in using the root filesystem for the data directory (not advised in most cases).

8. Enter **Y** or **N** for **Should this appliance run as a standalone database server?**
   - Select **Y** to configure the appliance as a database-only appliance. As a result, the appliance is configured as a basic PostgreSQL server, without a user interface.
   - Select **N** to configure the appliance with the full administrative user interface.
9. When prompted, enter a unique number to create a new region.

   Creating a new region destroys any existing data on the chosen database.
10. Create and confirm a password for the database.

Infrastructure Automation then configures the internal database. This takes a few minutes. After the database is created and initialized, you can log in to Infrastructure Automation.

Configuring an External Database

Based on your setup, you will choose to configure the appliance to use an external PostgreSQL database. For example, we can only have one database in a single region. However, a region can be segmented into multiple zones, such as database zone, user interface zone, and reporting zone, where each zone provides a specific function. The appliances in these zones must be configured to use an external database.

The **postgresql.conf** file requires specific settings for correct operation. For example, it must correctly reclaim table space, control session timeouts, and format the PostgreSQL server log for improved system support. It is recommended that external databases use a **postgresql.conf** file based on the standard file used by the Infrastructure Automation appliance.

Ensure you configure the settings in the **postgresql.conf** to suit your system. For example, customize the **shared_buffers** setting according to the amount of real storage available in the external system hosting the PostgreSQL instance. In addition, depending on the aggregate number of appliances expected to connect to the PostgreSQL instance, it may be necessary to alter the **max_connections** setting.

Note:

- Infrastructure Automation requires PostgreSQL version 13.
- **postgresql.conf** controls the operation of all databases managed by the PostgreSQL instance, therefore it is not recommended to run other databases on this PostgreSQL instance.

1. Start the appliance and open a terminal console.
2. Enter the **appliance_console** command. The appliance console summary screen will be displayed.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. Choose **Create Region in External Database** for the database location.
6. Enter the database hostname or IP address when prompted.
7. Enter the database name or leave blank for the default (**vmdb_production**).
8. Enter the database username or leave blank for the default (**root**).
9. Enter the chosen database user's password.
10. Confirm the configuration if prompted.

Infrastructure Automation will then configure the external database.

Configure Messaging

Configuring messaging is required for appliance setup. It is recommended to configure the broker on the same appliance where your database is configured

Note: You can only have one kafka broker per region

1. You can either configure the current appliance as a kafka broker, or point the appliance to an existing external kafka broker.

   Select the appropriate option either **Configure this appliance as a messaging server** or **Connect to an external messaging system** to connect to an external kafka broker. You will be asked to fill in the required **Message Client Parameters** like IP address and username/password.

2. Select **Proceed** and appliance_console will apply the configuration that you have requested then restart evmserverd to pick up the changes.

Configuring a Worker Appliance

You can use multiple appliances to facilitate horizontal scaling, as well as for dividing up work by roles. Accordingly, configure an appliance to handle work for one or many roles, with workers within the appliance carrying out the duties for which they are configured. You can configure a worker appliance through the terminal. The following steps demonstrate how to join a worker appliance to an appliance that already has a region configured with a database and messaging.

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The Infrastructure Automation appliance summary screen displays.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. You are prompted to create or fetch a security key. Since this is not the first Infrastructure Automation appliance, choose **2) Fetch key from remote machine**. For worker and multi-region setups, use this option to copy the security key from another appliance.

   Note:

   All Infrastructure Automation appliances in a multi-region deployment must use the same key.

6. Choose **Join Region in External Database** for the database location.
7. Enter the database hostname or IP address when prompted.
8. Enter the port number or leave blank for the default (**`5432`**).
9. Enter the database name or leave blank for the default (**`vmdb_production`**).
10. Enter the database username or leave blank for the default (**`root`**).
11. Enter the chosen database user's password.
12. Confirm the configuration if prompted.
13. Choose **Connect to an external messaging system** to connect to the external kafka broker located on the appliance with the external database

    Note: You can only have one kafka broker per region

14. Enter the necessary **Message Client Parameters** such as the hostname/IP and username/password
15. Confirm the configuration if prompted.

Once Infrastructure Automation is installed, you can log in and perform administration tasks.

Log in to Infrastructure Automation for the first time after installing by:

1. Browse to the URL for the login screen. (https://xx.xx.xx.xx on the virtual machine instance)

2. Enter the default credentials (Username: **admin** | Password: **smartvm**) for the initial login.
3. Click **Login**.

Appendix

Appliance Console Command-Line Interface (CLI)

Currently, the **appliance_console_cli** feature is a subset of the full functionality of the **appliance_console** itself, and covers functions most likely to be scripted by using the command-line interface (CLI).

1. After starting the Infrastructure Automation appliance, log in with a user name of **root** and the default password of **smartvm**. This displays the Bash prompt for the root user.
2. Enter the **appliance_console_cli** or **appliance_console_cli --help** command to see a list of options available with the command, or simply enter **appliance_console_cli --option <argument>** directly to use a specific option.

Database Configuration Options

Option	Description
–region (-r)	region number (create a new region in the database - requires database credentials passed)
–internal (-i)	internal database (create a database on the current appliance)
–dbdisk	database disk device path (for configuring an internal database)
–hostname (-h)	database hostname
–port	database port (defaults to **5432**)
–username (-U)	database username (defaults to **root**)
–password (-p)	database password
–dbname (-d)	database name (defaults to **vmdb_production**)

v2_key Options

Option	Description
–key (-k)	create a new v2_key
–fetch-key (-K)	fetch the v2_key from the given host
–force-key (-f)	create or fetch the key even if one exists
–sshlogin	ssh username for fetching the v2_key (defaults to **root**)
–sshpassword	ssh password for fetching the v2_key

IPA Server Options

Option	Description
–host (-H)	set the appliance hostname to the given name
–ipaserver (-e)	IPA server FQDN
–ipaprincipal (-n)	IPA server principal (default: **admin**)
–ipapassword (-w)	IPA server password
–ipadomain (-o)	IPA server domain (optional). Will be based on the appliance domain name if not specified.
–iparealm (-l)	IPA server realm (optional). Will be based on the domain name of the ipaserver if not specified.
–uninstall-ipa (-u)	uninstall IPA client

Note:

- In order to configure authentication through an IPA server, in addition to using **Configure External Authentication (httpd)** in the **`appliance_console`**, external authentication can be optionally configured via the **`appliance_console_cli`** (command-line interface).
- Specifying **–host** will update the hostname of the appliance. If this step was already performed via the **`appliance_console`** and the necessary updates that are made to **`/etc/hosts`** if DNS is not properly configured, the **–host** option can be omitted.

Certificate Options

Option	Description
–ca (-c)	CA name used for certmonger (default: **`ipa`**)
–postgres-client-cert (-g)	install certs for postgres client
–postgres-server-cert	install certs for postgres server
–http-cert	install certs for http server (to create certs/httpd* values for a unique key)
–extauth-opts (-x)	external authentication options

Note: The certificate options augment the functionality of the **`certmonger`** tool and enable creating a certificate signing request (CSR), and specifying **`certmonger`** the directories to store the keys.

Other Options

Option	Description
–logdisk (-l)	log disk path
–tmpdisk	initialize the given device for temp storage (volume mounted at **`/var/www/miq_tmp`**)
–verbose (-v)	print more debugging info

Example Usage.

```
$ ssh root@appliance.test.company.com
```

To create a new database locally on the server by using **`/dev/sdb`**:

```
# appliance_console_cli --internal --dbdisk /dev/sdb --region 0 --password smartvm
```

To copy the v2_key from a host *some.example.com* to local machine:

```
# appliance_console_cli --fetch-key some.example.com --sshlogin root --sshpassword smartvm
```

You could combine the two to join a region where *db.example.com* is the appliance hosting the database:

```
# appliance_console_cli --fetch-key db.example.com --sshlogin root --sshpassword smartvm --
hostname db.example.com --password mydatabasepassword
```

To configure external authentication:

```
# appliance_console_cli --host appliance.test.company.com
                        --ipaserver ipaserver.test.company.com
                        --ipadomain test.company.com
                        --iparealm TEST.COMPANY.COM
                        --ipaprincipal admin
                        --ipapassword smartvm1
```

To uninstall external authentication:

```
# appliance_console_cli  --uninstall-ipa
```


- Installing on VMware vSphere
  - Installing Infrastructure Automation
    - Obtaining the appliance
    - Uploading the Appliance on VMware vSphere
  - Configuring Infrastructure Automation

        - Accessing the Appliance Console
        - Configuring a Database
            - Configuring an Internal Database
            - Configuring an External Database
        - Configure Messaging
        - Configuring a Worker Appliance
    - Additional Configuration for Appliances on VMware vSphere
        - Tuning Appliance Performance
    - Logging In After Installing Infrastructure Automation
        - Changing the Default Login Password
- Appendix
- Appliance Console Command-Line Interface (CLI)
    - Database Configuration Options
    - v2_key Options
    - IPA Server Options
    - Certificate Options
    - Other Options

Installing on VMware vSphere

Installing Infrastructure Automation

Installing Infrastructure Automation consists of the following steps:

1. Downloading the appliance for your environment as a virtual machine image template.
2. Setting up a virtual machine based on the appliance.
3. Configuring the Infrastructure Automation appliance.

After you have completed all the procedures in this guide, you will have a working environment on which additional customizations and configurations can be performed.

Obtaining the appliance

Uploading the Appliance on VMware vSphere

Uploading the Infrastructure Automation appliance file onto VMware vSphere systems has the following requirements:

- 44 GB of space on the chosen vSphere datastore.
- 12 GB RAM.
- 4 VCPUs.
- Administrator access to the vSphere Client.
- Depending on your infrastructure, allow time for the upload.

Note:

These are the procedural steps as of the time of writing. For more information, consult the “Deploying an OVF or OVA Template” topic in the VMware documentation.

Use the following procedure to upload the Infrastructure Automation appliance OVF template from your local file system using the vSphere Client.

1. In the vSphere Client, select **File > Deploy OVF Template**. The Deploy OVF Template wizard appears.
2. Specify the source location and click **Next**.

- Select **Deploy from File** to browse your file system for the OVF template, for example **`manageiq-vsphere-ivanchuk-4.ova`**.
- Select **Deploy from URL** to specify a URL to an OVF template located on the internet.

3. View the **OVF Template Details** page and click **Next**.
4. Select the deployment configuration from the drop-down menu and click **Next**. The option selected typically controls the memory settings, number of CPUs and reservations, and application-level configuration parameters.
5. Select the host or cluster on which you want to deploy the OVF template and click **Next**.
6. Select the host on which you want to run the appliance, and click **Next**.
7. Navigate to, and select the resource pool where you want to run the Infrastructure Automation appliance and click **Next**.
8. Select a datastore to store the deployed Infrastructure Automation appliance, and click Next. Ensure to select a datastore large enough to accommodate the virtual machine and all of its virtual disk files.
9. Select the disk format to store the virtual machine virtual disks, and click **Next**.
   - Select **Thin Provisioned** if the storage is allocated on demand as data is written to the virtual disks.
   - Select **Thick Provisioned** if all storage is immediately allocated.
10. For each network specified in the OVF template, select a network by right-clicking the **Destination Network** column in your infrastructure to set up the network mapping and click **Next**.
11. The **IP Allocation** page does not require any configuration changes. Leave the default settings in the **IP Allocation** page and click **Next**.
12. Set the user-configurable properties and click **Next**. The properties to enter depend on the selected IP allocation scheme. For example, you are prompted for IP related information for the deployed virtual machines only in the case of a fixed IP allocation scheme.
13. Review your settings and click **Finish**.

The progress of the import task appears in the vSphere Client Status panel.

Configuring Infrastructure Automation

After installing Infrastructure Automation and running it for the first time, you must perform some basic configuration. You must complete these steps:

1. Add a disk to the infrastructure that is hosting your appliance.
2. Configure the database.
3. Configure messaging

Configure the appliance by using the internal appliance console.

Accessing the Appliance Console

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The Infrastructure Automation appliance summary screen displays.
3. Press **`Enter`** to manually configure settings.
4. Press the number for the item you want to change, and press **`Enter`**. The options for your selection are displayed.
5. Follow the prompts to make the changes.
6. Press **`Enter`** to accept a setting where applicable.

Note:

The Infrastructure Automation appliance console automatically logs out after five minutes of inactivity.

Configuring a Database

Infrastructure Automation uses a database to store information about the environment. Before using Infrastructure Automation, configure the database options for it; Infrastructure Automation provides the following two options for database configuration:

Configuring an Internal Database

Before installing an internal database, add a disk to the infrastructure hosting your appliance. See the documentation specific to your infrastructure for instructions for adding a disk. As a storage disk usually cannot be added while a virtual machine is running, Red Hat recommends adding the disk before starting the appliance. Infrastructure Automation only supports installing of an internal VMDB on blank disks; installation will fail if the disks are not blank.

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The Infrastructure Automation appliance summary screen displays.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. You are prompted to create or fetch an encryption key.
    - If this is the first Infrastructure Automation appliance, choose **Create key**.
    - If this is not the first Infrastructure Automation appliance, choose **Fetch key from remote machine** to fetch the key from the first appliance. For worker and multi-region setups, use this option to copy key from another appliance.

      Note:

      All Infrastructure Automation appliances in a multi-region deployment must use the same key.
6. Choose **Create Internal Database** for the database location.

   In the **Configure Messaging** menu, select **Make No messaging changes**. If you see **`Configuration failed: Internal database require a volume mounted at /var/lib/pgsql. Please add an unpartitioned disk and try again.`** message, then ensure to add a second disk for the database per instructions as defined above.
7. Choose a disk for the database. This can be either a disk you attached previously, or a partition on the current disk.

   Red Hat recommends using a separate disk for the database.
   If there is an unpartitioned disk attached to the virtual machine, the dialog will show options similar to the following:

   ```
   1) /dev/vdb: 20480
   2) Don't partition the disk
   ```

    - Enter **1** to choose **`/dev/vdb`** for the database location. This option creates a logical volume using this device and mounts the volume to the appliance in a location appropriate for storing the database. The default location is **`/var/lib/pgsql`**, which can be found in the environment variable **`$APPLIANCE_PG_MOUNT_POINT`**.
    - Enter **2** to continue without partitioning the disk. A second prompt will confirm this choice. Selecting this option results in using the root filesystem for the data directory (not advised in most cases).
8. Enter **Y** or **N** for **Should this appliance run as a standalone database server?**
    - Select **Y** to configure the appliance as a database-only appliance. As a result, the appliance is configured as a basic PostgreSQL server, without a user interface.
    - Select **N** to configure the appliance with the full administrative user interface.

9. When prompted, enter a unique number to create a new region.

   Creating a new region destroys any existing data on the chosen database.
10. Create and confirm a password for the database.

Infrastructure Automation then configures the internal database. This takes a few minutes. After the database is created and initialized, you can log in to Infrastructure Automation.

Configuring an External Database

Based on your setup, you will choose to configure the appliance to use an external PostgreSQL database. For example, we can only have one database in a single region. However, a region can be segmented into multiple zones, such as database zone, user interface zone, and reporting zone, where each zone provides a specific function. The appliances in these zones must be configured to use an external database.

The **`postgresql.conf`** file requires specific settings for correct operation. For example, it must correctly reclaim table space, control session timeouts, and format the PostgreSQL server log for improved system support. It is recommended that external databases use a **`postgresql.conf`** file based on the standard file used by the Infrastructure Automation appliance.

Ensure you configure the settings in the **`postgresql.conf`** to suit your system. For example, customize the **`shared_buffers`** setting according to the amount of real storage available in the external system hosting the PostgreSQL instance. In addition, depending on the aggregate number of appliances expected to connect to the PostgreSQL instance, it may be necessary to alter the **`max_connections`** setting.

Note:

- Infrastructure Automation requires PostgreSQL version 13.
- **`postgresql.conf`** controls the operation of all databases managed by the PostgreSQL instance, therefore it is not recommended to run other databases on this PostgreSQL instance.

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The appliance console summary screen will be displayed.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. Choose **Create Region in External Database** for the database location.
6. Enter the database hostname or IP address when prompted.
7. Enter the database name or leave blank for the default (**`vmdb_production`**).
8. Enter the database username or leave blank for the default (**`root`**).
9. Enter the chosen database user's password.
10. Confirm the configuration if prompted.

Infrastructure Automation will then configure the external database.

Configure Messaging

Configuring messaging is required for appliance setup. It is recommended to configure the broker on the same appliance where your database is configured

Note: You can only have one kafka broker per region

1. You can either configure the current appliance as a kafka broker, or point the appliance to an existing external kafka broker.

   Select the appropriate option either **Configure this appliance as a messaging server** or **Connect to an external messaging system** to connect to an external kafka broker. You will be asked to fill in the required **Message Client Parameters** like IP address and username/password.

2. Select **Proceed** and appliance_console will apply the configuration that you have requested then restart evmserverd to pick up the changes.

Configuring a Worker Appliance

You can use multiple appliances to facilitate horizontal scaling, as well as for dividing up work by roles. Accordingly, configure an appliance to handle work for one or many roles, with workers within the appliance carrying out the duties for which they are configured. You can configure a worker appliance through the terminal. The following steps demonstrate how to join a worker appliance to an appliance that already has a region configured with a database and messaging.

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The Infrastructure Automation appliance summary screen displays.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. You are prompted to create or fetch a security key. Since this is not the first Infrastructure Automation appliance, choose **2) Fetch key from remote machine**. For worker and multi-region setups, use this option to copy the security key from another appliance.

   Note:

   All Infrastructure Automation appliances in a multi-region deployment must use the same key.
6. Choose **Join Region in External Database** for the database location.
7. Enter the database hostname or IP address when prompted.
8. Enter the port number or leave blank for the default (**`5432`**).
9. Enter the database name or leave blank for the default (**`vmdb_production`**).
10. Enter the database username or leave blank for the default (**`root`**).
11. Enter the chosen database user's password.
12. Confirm the configuration if prompted.
13. Choose **Connect to an external messaging system** to connect to the external kafka broker located on the appliance with the external database

    Note: You can only have one kafka broker per region
14. Enter the necessary **Message Client Parameters** such as the hostname/IP and username/password
15. Confirm the configuration if prompted.

Additional Configuration for Appliances on VMware vSphere

Tuning Appliance Performance

By default, the Infrastructure Automation appliance uses the **`tuned`** service and its **`virtual-guest`** profile to optimize performance. In most cases, this profile provides the best performance for the appliance.

However on some VMware setups (for example, with a large vCenter database), the following additional tuning may further improve appliance performance:

- When using the **`virtual-guest`** profile in **`tuned`**, edit the **`vm.swappiness`** setting to **`1`** in the **`tuned.conf`** file from the default of **`vm.swappiness = 30`**.
- Use the **`noop`** scheduler instead. See the VMware documentation for more details on the best scheduler for your environment. See Setting the Default I/O Scheduler in the Red Hat Enterprise Linux *Performance Tuning Guide* for instructions on changing the default I/O scheduler.

Logging In After Installing Infrastructure Automation

Once Infrastructure Automation is installed, you can log in and perform administration tasks.

Log in to Infrastructure Automation for the first time after installing by:

1. Browse to the URL for the login screen. (https://xx.xx.xx.xx on the virtual machine instance)
2. Enter the default credentials (Username: **admin** | Password: **smartvm**) for the initial login.
3. Click **Login**.

Changing the Default Login Password

Change your password to ensure more private and secure access to Infrastructure Automation.

1. Browse to the URL for the login screen. (https://xx.xx.xx.xx on the virtual machine instance)
2. Click **Update Password** beneath the **Username** and **Password** text fields.
3. Enter your current **Username** and **Password** in the text fields.
4. Input a new password in the **New Password** field.
5. Repeat your new password in the **Verify Password** field.
6. Click **Login**.

Appendix

Appliance Console Command-Line Interface (CLI)

Currently, the **`appliance_console_cli`** feature is a subset of the full functionality of the **`appliance_console`** itself, and covers functions most likely to be scripted by using the command-line interface (CLI).

1. After starting the Infrastructure Automation appliance, log in with a user name of **`root`** and the default password of **`smartvm`**. This displays the Bash prompt for the root user.
2. Enter the **`appliance_console_cli`** or **`appliance_console_cli --help`** command to see a list of options available with the command, or simply enter **`appliance_console_cli --option <argument>`** directly to use a specific option.

Database Configuration Options

Option	Description
–region (-r)	region number (create a new region in the database - requires database credentials passed)
–internal (-i)	internal database (create a database on the current appliance)
–dbdisk	database disk device path (for configuring an internal database)
–hostname (-h)	database hostname
–port	database port (defaults to **`5432`**)
–username (-U)	database username (defaults to **`root`**)
–password (-p)	database password
–dbname (-d)	database name (defaults to **`vmdb_production`**)

v2_key Options

Option	Description
–key (-k)	create a new v2_key
–fetch-key (-K)	fetch the v2_key from the given host
–force-key (-f)	create or fetch the key even if one exists
–sshlogin	ssh username for fetching the v2_key (defaults to **root**)
–sshpassword	ssh password for fetching the v2_key

IPA Server Options

Option	Description
–host (-H)	set the appliance hostname to the given name
–ipaserver (-e)	IPA server FQDN
–ipaprincipal (-n)	IPA server principal (default: **admin**)
–ipapassword (-w)	IPA server password
–ipadomain (-o)	IPA server domain (optional). Will be based on the appliance domain name if not specified.
–iparealm (-l)	IPA server realm (optional). Will be based on the domain name of the ipaserver if not specified.
–uninstall-ipa (-u)	uninstall IPA client

Note:

- In order to configure authentication through an IPA server, in addition to using **Configure External Authentication (httpd)** in the **appliance_console**, external authentication can be optionally configured via the **appliance_console_cli** (command-line interface).
- Specifying **–host** will update the hostname of the appliance. If this step was already performed via the **appliance_console** and the necessary updates that are made to **/etc/hosts** if DNS is not properly configured, the **–host** option can be omitted.

Certificate Options

Option	Description
–ca (-c)	CA name used for certmonger (default: **ipa**)
–postgres-client-cert (-g)	install certs for postgres client
–postgres-server-cert	install certs for postgres server
–http-cert	install certs for http server (to create certs/httpd* values for a unique key)
–extauth-opts (-x)	external authentication options

Note: The certificate options augment the functionality of the **certmonger** tool and enable creating a certificate signing request (CSR), and specifying **certmonger** the directories to store the keys.

Other Options

Option	Description
–logdisk (-l)	log disk path
–tmpdisk	initialize the given device for temp storage (volume mounted at **/var/www/miq_tmp**)
–verbose (-v)	print more debugging info

Example Usage.

```
$ ssh root@appliance.test.company.com
```

To create a new database locally on the server by using **/dev/sdb**:

```
# appliance_console_cli --internal --dbdisk /dev/sdb --region 0 --password smartvm
```

To copy the v2_key from a host *some.example.com* to local machine:

```
# appliance_console_cli --fetch-key some.example.com --sshlogin root --sshpassword smartvm
```

You could combine the two to join a region where *db.example.com* is the appliance hosting the database:

```
# appliance_console_cli --fetch-key db.example.com --sshlogin root --sshpassword smartvm --
hostname db.example.com --password mydatabasepassword
```

To configure external authentication:

```
# appliance_console_cli --host appliance.test.company.com
                        --ipaserver ipaserver.test.company.com
                        --ipadomain test.company.com
                        --iparealm TEST.COMPANY.COM
                        --ipaprincipal admin
                        --ipapassword smartvm1
```

To uninstall external authentication:

```
# appliance_console_cli  --uninstall-ipa
```


- Installing on Red Hat OpenStack Platform
  - Installing Infrastructure Automation
    - Obtaining the appliance
    - Uploading the Appliance on OpenStack
    - Adding a Rule to a Security Group
    - Creating a Custom Flavor
    - Launching the Instance
    - Adding a Floating IP Address
  - Configuring Infrastructure Automation
    - Accessing the Appliance Console
    - Configuring a Database
      - Configuring an Internal Database
      - Configuring an External Database
    - Configure Messaging
    - Configuring a Worker Appliance
  - Logging In After Installing Infrastructure Automation
    - Changing the Default Login Password
- Appendix
- Appliance Console Command-Line Interface (CLI)
  - Database Configuration Options
  - v2_key Options
  - IPA Server Options
  - Certificate Options
  - Other Options

Installing on Red Hat OpenStack Platform

Installing Infrastructure Automation

Installing Infrastructure Automation consists of the following steps:

1. Downloading the appliance for your environment as a virtual machine image template.
2. Setting up a virtual machine based on the appliance.
3. Configuring the Infrastructure Automation appliance.

After you have completed all the procedures in this guide, you will have a working environment on which additional customizations and configurations can be performed.

Obtaining the appliance

Uploading the Appliance on OpenStack

Log in to your OpenStack dashboard to upload your Infrastructure Automation appliance.

1. Log in to the OpenStack dashboard.
2. In the **Project** tab, browse to menu: **Compute > Images**.
3. Click **Create Image**.
4. In **Name**, enter a name for the image.
5. From **Image Source** list, select **Image Location**. Note that currently only images available via an HTTP URL are supported.
6. In **Image Location**, add an external (HTTP) URL to load the image from. For example, **http://example.com/image.iso**.
7. From the **Format** list, select the image format. For example, `ISO
   - Optical Disk Image`.
8. Specify the **Architecture**. For example, **`i386`** for a 32-bit architecture or **`x86-64`** for a 64-bit architecture.
9. Leave the **Minimum Disk (GB)** and **Minimum RAM (MB)** fields empty.
10. Check the **Public** box to make the appliance available to all users.
11. Check the **Protected** box to protect the image from being accidentally deleted.
12. Click **Create Image**.

You have successfully uploaded the Infrastructure Automation appliance.

The appliance image is placed in a queue to be uploaded. It may take some time before the Status of the image changes from Queued to Active.

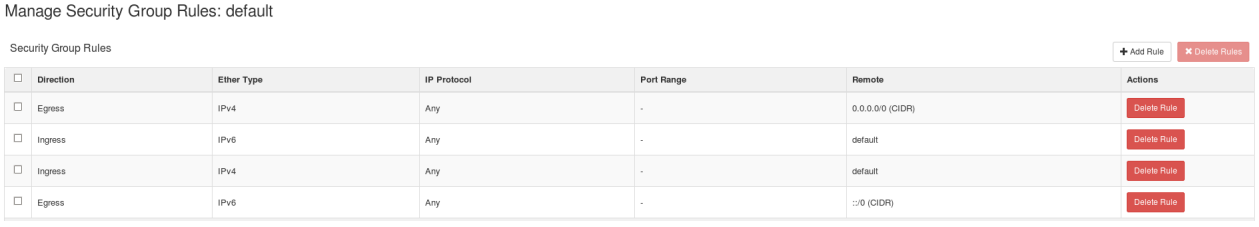
Adding a Rule to a Security Group

Security groups specify what IP traffic is allowed to reach an instance on its public IP address. Security group rules are processed before network traffic reaches firewall rules defined within the guest itself.

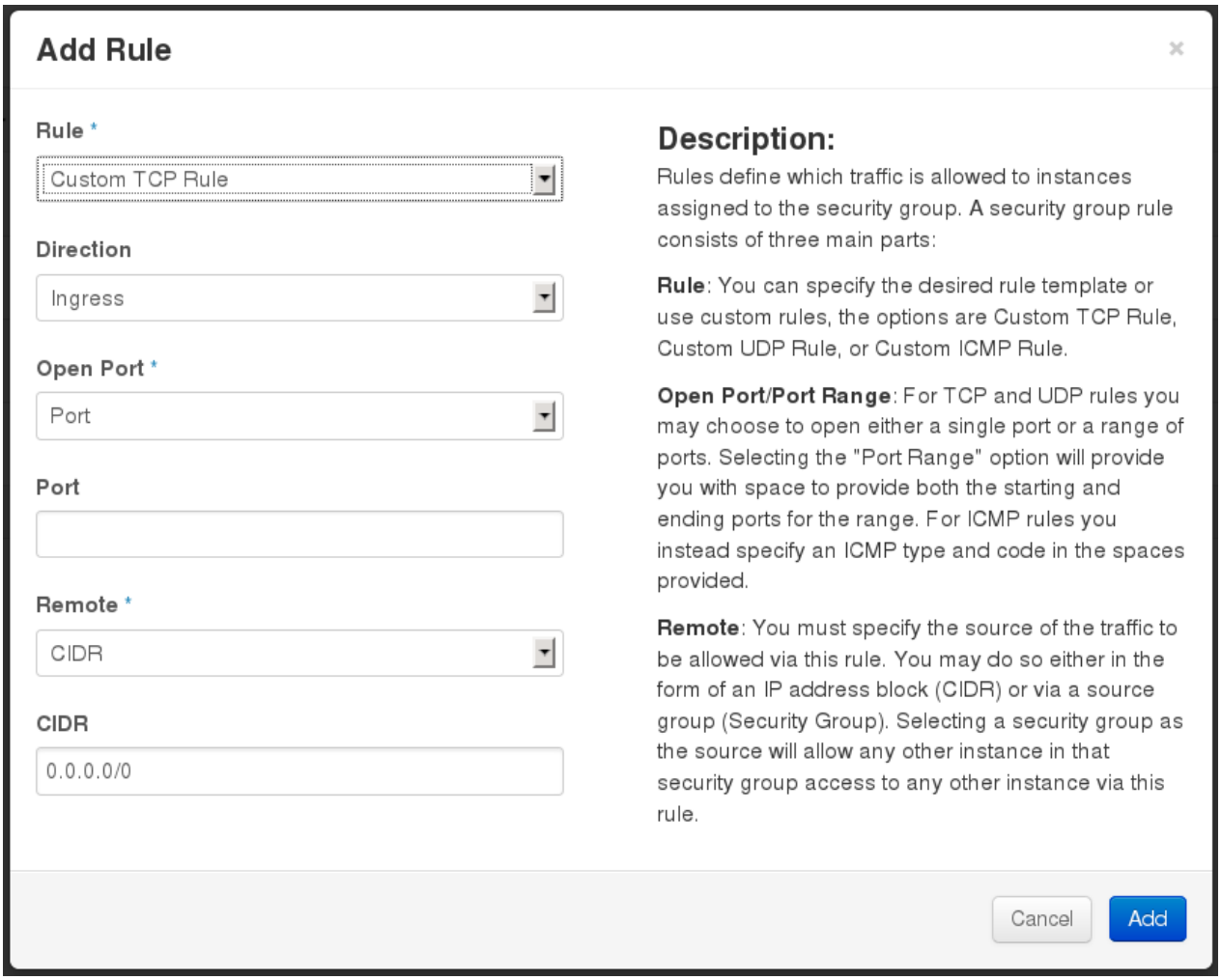
Note:

In the default configuration, the default security group accepts all connections from the default source; all instances within the default group can talk to each other on any port.

1. From the OpenStack dashboard, browse to menu: **Project > Compute > Access & Security**.
2. Browse to menu: **Security Groups > Manage Rules on the row for the default security group.



3. Click **Add Rule**.



4. Configure the rule.
    1. Select **Rule > Custom TCP Rule**.
    2. Select **Direction > Ingress**.
    3. Select **Port** from the **Open Port** list.
    4. Specify `443` in the **Port** field.
    5. Select **CIDR** from the **Remote** list.
    6. Specify `0.0.0.0/0` in the **CIDR** field.
    7. Click **Add**.

Creating a Custom Flavor

A flavor is a resource allocation profile that specifies, for example, how many virtual CPUs and how much RAM can be allocated to an instance. You can, for example, run Infrastructure Automation on a Red Hat OpenStack m1.large flavor, which specifies a virtual machine with 4 cores, 12 GB RAM, and 80 GB disk space. Creating a flavor to run Infrastructure Automation is optional.

The following procedure demonstrates creating a flavor with the minimum requirements (4 cores, 12 GB RAM, 44 GB disk space) for Infrastructure Automation. For more information about flavors, see the Red Hat OpenStack Platform Administration User Guide.

1. Log in to the OpenStack dashboard as admin.
2. In the **Admin** tab, browse to menu: **System > Flavors**.

3. Click **Create Flavor** to display the **Create Flavor** dialog.
4. Configure the settings to define a flavor that meets Infrastructure Automation system requirements.
    1. Enter a name for the flavor.
    2. Enter the following settings:
        - **VCPUs**: 4
        - **RAM MB**: 12288
        - **Root Disk GB**: 70
        - **Ephemeral Disk GB**: 0
        - **Swap Disk MB**: 0
5. Click **Create Flavor**.

A new flavor specific to Infrastructure Automation is created.

Launching the Instance

1. From the OpenStack dashboard, browse to menu: **Project > Compute > Instances**.
2. Click **Launch Instance**.
3. Enter a name for the instance.
4. Select the custom flavor for your instance. The flavor selection determines the computing resources available to your instance. The resources used by the flavor are displayed in the **Flavor Details** pane.
5. Enter **1** in the **Instance Count** field.
6. Select a boot option from the **Instance Boot Source** list:
    - **Boot from image** - displays a new field for **Image Name**. Select the image from the drop-down list.
    - **Boot from snapshot** - displays a new field for **Instance Snapshot**. Select the snapshot from the drop-down list.
    - **Boot from volume** - displays a new field for **Volume**. Select the volume from the drop-down list.
    - **Boot from image (creates a new volume)** - boot from an image and create a volume by choosing **Device Size** and **Device Name** for your volume. Some volumes can be persistent. To ensure the volume is deleted when the instance is deleted, select **Delete on Terminate**.
    - **Boot from volume snapshot (creates a new volume)** - boot from volume snapshot and create a new volume by choosing **Volume Snapshot** from the drop-down list and adding a **Device Name** for your volume. Some volumes can be persistent. To ensure the volume is deleted when the instance is deleted, select **Delete on Terminate**.
7. Click **Networking** and select a network for the instance by clicking the **+** (plus) button for the network from **Available Networks**.
8. Click **Launch**.

Adding a Floating IP Address

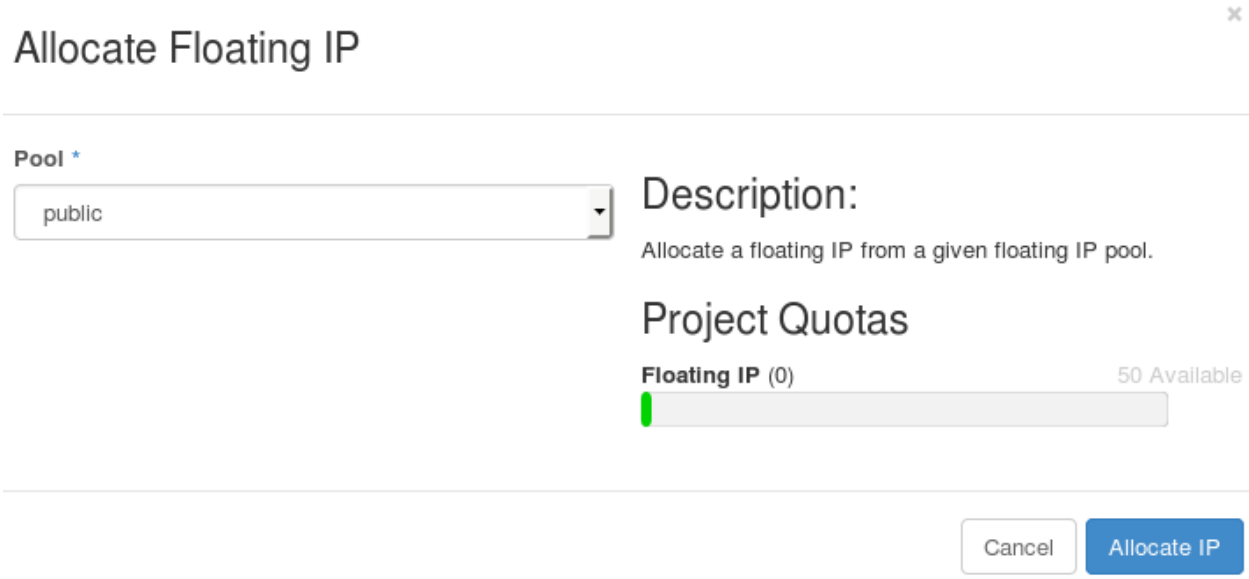
When you create an instance, Red Hat OpenStack Platform automatically assigns it a fixed IP address in the network to which the instance belongs. This IP address is permanently associated with the instance until the instance is terminated.

In addition to the fixed address, you can also assign a floating IP address to an instance. Unlike fixed IP addresses, you can modify floating IP addresses associations at any time, regardless of the state of the instances involved.

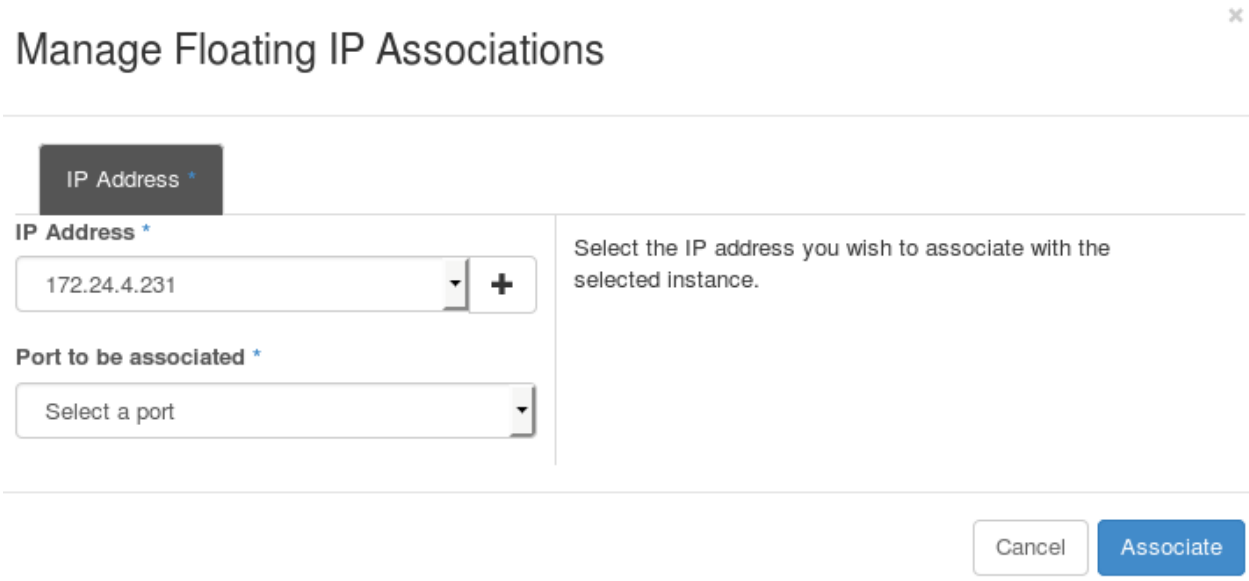
1. At the command-line on your RHEL OpenStack Platform host, create a pool of floating IP addresses using the **`nova-manage floating create`** command. Replace **IP_BLOCK** with the desired block of IP addresses expressed in CIDR notation.

```
$ nova-manage floating create IP_BLOCK
```

2. In the **Project** tab, browse to **Compute > Access & Security**.
3. Click **Floating IPs > Allocate IP To Project**. The **Allocate Floating IP** window is displayed.



4. Click **Allocate IP** to allocate a floating IP from the pool. The allocated IP address appears in the **Floating IPs** table.
5. Select the newly allocated IP address from the **Floating IPs** table. Click **Associate** to assign the IP address to a specific instance.



6. Select an instance with which to associate the floating IP Address.
7. Click **Associate** to associate the IP address with the selected instance.

Note:

To disassociate a floating IP address from an instance when it is no longer required, click **Release Floating IPs**.

Configuring Infrastructure Automation

After installing Infrastructure Automation and running it for the first time, you must perform some basic configuration. You must complete these steps:

1. Add a disk to the infrastructure that is hosting your appliance.
2. Configure the database.
3. Configure messaging

Configure the appliance by using the internal appliance console.

Accessing the Appliance Console

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The Infrastructure Automation appliance summary screen displays.
3. Press **`Enter`** to manually configure settings.
4. Press the number for the item you want to change, and press **`Enter`**. The options for your selection are displayed.
5. Follow the prompts to make the changes.
6. Press **`Enter`** to accept a setting where applicable.

Note:

The Infrastructure Automation appliance console automatically logs out after five minutes of inactivity.

Configuring a Database

Infrastructure Automation uses a database to store information about the environment. Before using Infrastructure Automation, configure the database options for it; Infrastructure Automation provides the following two options for database configuration:

Configuring an Internal Database

Before installing an internal database, add a disk to the infrastructure hosting your appliance. See the documentation specific to your infrastructure for instructions for adding a disk. As a storage disk usually cannot be added while a virtual machine is running, Red Hat recommends adding the disk before starting the appliance. Infrastructure Automation only supports installing of an internal VMDB on blank disks; installation will fail if the disks are not blank.

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The Infrastructure Automation appliance summary screen displays.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. You are prompted to create or fetch an encryption key.
   - If this is the first Infrastructure Automation appliance, choose **Create key**.
   - If this is not the first Infrastructure Automation appliance, choose **Fetch key from remote machine** to fetch the key from the first appliance. For worker and multi-region setups, use this option to copy key from another appliance.

     Note:

     All Infrastructure Automation appliances in a multi-region deployment must use the same key.
6. Choose **Create Internal Database** for the database location.

   In the **Configure Messaging** menu, select **Make No messaging changes**. If you see **`Configuration failed: Internal database require a volume mounted at /var/lib/pgsql. Please add an unpartitioned disk and try again.`** message, then ensure to add a second disk for the database per instructions as defined above.

7. Choose a disk for the database. This can be either a disk you attached previously, or a partition on the current disk.

   Red Hat recommends using a separate disk for the database.
   If there is an unpartitioned disk attached to the virtual machine, the dialog will show options similar to the following:

   ```
   1) /dev/vdb: 20480
   2) Don't partition the disk
   ```

   - Enter **1** to choose **/dev/vdb** for the database location. This option creates a logical volume using this device and mounts the volume to the appliance in a location appropriate for storing the database. The default location is **/var/lib/pgsql**, which can be found in the environment variable **$APPLIANCE_PG_MOUNT_POINT**.
   - Enter **2** to continue without partitioning the disk. A second prompt will confirm this choice. Selecting this option results in using the root filesystem for the data directory (not advised in most cases).

8. Enter **Y** or **N** for **Should this appliance run as a standalone database server?**

   - Select **Y** to configure the appliance as a database-only appliance. As a result, the appliance is configured as a basic PostgreSQL server, without a user interface.
   - Select **N** to configure the appliance with the full administrative user interface.

9. When prompted, enter a unique number to create a new region.

   Creating a new region destroys any existing data on the chosen database.
10. Create and confirm a password for the database.

Infrastructure Automation then configures the internal database. This takes a few minutes. After the database is created and initialized, you can log in to Infrastructure Automation.

Configuring an External Database

Based on your setup, you will choose to configure the appliance to use an external PostgreSQL database. For example, we can only have one database in a single region. However, a region can be segmented into multiple zones, such as database zone, user interface zone, and reporting zone, where each zone provides a specific function. The appliances in these zones must be configured to use an external database.

The **postgresql.conf** file requires specific settings for correct operation. For example, it must correctly reclaim table space, control session timeouts, and format the PostgreSQL server log for improved system support. It is recommended that external databases use a **postgresql.conf** file based on the standard file used by the Infrastructure Automation appliance.

Ensure you configure the settings in the **postgresql.conf** to suit your system. For example, customize the **shared_buffers** setting according to the amount of real storage available in the external system hosting the PostgreSQL instance. In addition, depending on the aggregate number of appliances expected to connect to the PostgreSQL instance, it may be necessary to alter the **max_connections** setting.

Note:

- Infrastructure Automation requires PostgreSQL version 13.
- **postgresql.conf** controls the operation of all databases managed by the PostgreSQL instance, therefore it is not recommended to run other databases on this PostgreSQL instance.

1. Start the appliance and open a terminal console.
2. Enter the **appliance_console** command. The appliance console summary screen will be displayed.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. Choose **Create Region in External Database** for the database location.
6. Enter the database hostname or IP address when prompted.
7. Enter the database name or leave blank for the default (**vmdb_production**).

8. Enter the database username or leave blank for the default (**root**).
9. Enter the chosen database user's password.
10. Confirm the configuration if prompted.

Infrastructure Automation will then configure the external database.

Configure Messaging

Configuring messaging is required for appliance setup. It is recommended to configure the broker on the same appliance where your database is configured

Note: You can only have one kafka broker per region

1. You can either configure the current appliance as a kafka broker, or point the appliance to an existing external kafka broker.

   Select the appropriate option either **Configure this appliance as a messaging server** or **Connect to an external messaging system** to connect to an external kafka broker. You will be asked to fill in the required **Message Client Parameters** like IP address and username/password.

2. Select **Proceed** and appliance_console will apply the configuration that you have requested then restart evmserverd to pick up the changes.

Configuring a Worker Appliance

You can use multiple appliances to facilitate horizontal scaling, as well as for dividing up work by roles. Accordingly, configure an appliance to handle work for one or many roles, with workers within the appliance carrying out the duties for which they are configured. You can configure a worker appliance through the terminal. The following steps demonstrate how to join a worker appliance to an appliance that already has a region configured with a database and messaging.

1. Start the appliance and open a terminal console.
2. Enter the **appliance_console** command. The Infrastructure Automation appliance summary screen displays.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. You are prompted to create or fetch a security key. Since this is not the first Infrastructure Automation appliance, choose **2) Fetch key from remote machine**. For worker and multi-region setups, use this option to copy the security key from another appliance.

   Note:

   All Infrastructure Automation appliances in a multi-region deployment must use the same key.

6. Choose **Join Region in External Database** for the database location.
7. Enter the database hostname or IP address when prompted.
8. Enter the port number or leave blank for the default (**5432**).
9. Enter the database name or leave blank for the default (**vmdb_production**).
10. Enter the database username or leave blank for the default (**root**).
11. Enter the chosen database user's password.
12. Confirm the configuration if prompted.
13. Choose **Connect to an external messaging system** to connect to the external kafka broker located on the appliance with the external database

    Note: You can only have one kafka broker per region

14. Enter the necessary **Message Client Parameters** such as the hostname/IP and username/password
15. Confirm the configuration if prompted.

Logging In After Installing Infrastructure Automation

Once Infrastructure Automation is installed, you can log in and perform administration tasks.

Log in to Infrastructure Automation for the first time after installing by:

1. Browse to the URL for the login screen. (https://xx.xx.xx.xx on the virtual machine instance)
2. Enter the default credentials (Username: **admin** | Password: **smartvm**) for the initial login.
3. Click **Login**.

Changing the Default Login Password

Change your password to ensure more private and secure access to Infrastructure Automation.

1. Browse to the URL for the login screen. (https://xx.xx.xx.xx on the virtual machine instance)
2. Click **Update Password** beneath the **Username** and **Password** text fields.
3. Enter your current **Username** and **Password** in the text fields.
4. Input a new password in the **New Password** field.
5. Repeat your new password in the **Verify Password** field.
6. Click **Login**.

Appendix

Appliance Console Command-Line Interface (CLI)

Currently, the **`appliance_console_cli`** feature is a subset of the full functionality of the **`appliance_console`** itself, and covers functions most likely to be scripted by using the command-line interface (CLI).

1. After starting the Infrastructure Automation appliance, log in with a user name of **`root`** and the default password of **`smartvm`**. This displays the Bash prompt for the root user.
2. Enter the **`appliance_console_cli`** or **`appliance_console_cli --help`** command to see a list of options available with the command, or simply enter **`appliance_console_cli --option <argument>`** directly to use a specific option.

Database Configuration Options

Option	Description
–region (-r)	region number (create a new region in the database - requires database credentials passed)
–internal (-i)	internal database (create a database on the current appliance)
–dbdisk	database disk device path (for configuring an internal database)
–hostname (-h)	database hostname
–port	database port (defaults to **`5432`**)
–username (-U)	database username (defaults to **`root`**)
–password (-p)	database password

–dbname (-d)	database name (defaults to `vmdb_production`)

v2_key Options

Option	Description
–key (-k)	create a new v2_key
–fetch-key (-K)	fetch the v2_key from the given host
–force-key (-f)	create or fetch the key even if one exists
–sshlogin	ssh username for fetching the v2_key (defaults to `root`)
–sshpassword	ssh password for fetching the v2_key

IPA Server Options

Option	Description
–host (-H)	set the appliance hostname to the given name
–ipaserver (-e)	IPA server FQDN
–ipaprincipal (-n)	IPA server principal (default: `admin`)
–ipapassword (-w)	IPA server password
–ipadomain (-o)	IPA server domain (optional). Will be based on the appliance domain name if not specified.
–iparealm (-l)	IPA server realm (optional). Will be based on the domain name of the ipaserver if not specified.
–uninstall-ipa (-u)	uninstall IPA client

Note:

- In order to configure authentication through an IPA server, in addition to using **Configure External Authentication (httpd)** in the `appliance_console`, external authentication can be optionally configured via the `appliance_console_cli` (command-line interface).
- Specifying **–host** will update the hostname of the appliance. If this step was already performed via the `appliance_console` and the necessary updates that are made to `/etc/hosts` if DNS is not properly configured, the **–host** option can be omitted.

Certificate Options

Option	Description
–ca (-c)	CA name used for certmonger (default: `ipa`)
–postgres-client-cert (-g)	install certs for postgres client
–postgres-server-cert	install certs for postgres server
–http-cert	install certs for http server (to create certs/httpd* values for a unique key)
–extauth-opts (-x)	external authentication options

Note: The certificate options augment the functionality of the `certmonger` tool and enable creating a certificate signing request (CSR), and specifying `certmonger` the directories to store the keys.

Other Options

Option	Description
–logdisk (-l)	log disk path
–tmpdisk	initialize the given device for temp storage (volume mounted at `/var/www/miq_tmp`)
–verbose (-v)	print more debugging info

Example Usage.

```
$ ssh root@appliance.test.company.com
```

To create a new database locally on the server by using **/dev/sdb**:

```
# appliance_console_cli --internal --dbdisk /dev/sdb --region 0 --password smartvm
```

To copy the v2_key from a host *some.example.com* to local machine:

```
# appliance_console_cli --fetch-key some.example.com --sshlogin root --sshpassword smartvm
```

You could combine the two to join a region where *db.example.com* is the appliance hosting the database:

```
# appliance_console_cli --fetch-key db.example.com --sshlogin root --sshpassword smartvm --
hostname db.example.com --password mydatabasepassword
```

To configure external authentication:

```
# appliance_console_cli --host appliance.test.company.com
                        --ipaserver ipaserver.test.company.com
                        --ipadomain test.company.com
                        --iparealm TEST.COMPANY.COM
                        --ipaprincipal admin
                        --ipapassword smartvm1
```

To uninstall external authentication:

```
# appliance_console_cli  --uninstall-ipa
```


- Installing on Red Hat Virtualization
  - Installing Infrastructure Automation
    - Obtaining the Appliance
    - Uploading the Appliance to Red Hat Virtualization
    - Creating a Infrastructure Automation Virtual Machine
  - Configuring Infrastructure Automation
    - Accessing the Appliance Console
    - Configuring a Database
      - Configuring an Internal Database
      - Configuring an External Database
    - Configure Messaging
    - Configuring a Worker Appliance
  - Logging In After Installing Infrastructure Automation
    - Changing the Default Login Password
- Appendix
- Appliance Console Command-Line Interface (CLI)
  - Database Configuration Options
  - v2_key Options
  - IPA Server Options
  - Certificate Options
  - Other Options

Installing on Red Hat Virtualization

Installing Infrastructure Automation

Installing Infrastructure Automation on Red Hat Virtualization consists of the following steps:

1. Downloading the appliance for your environment as a virtual machine image template.
2. Uploading the appliance image to the Red Hat Virtualization data storage domain.
3. Creating a virtual machine based on the appliance.

After you have completed all the procedures in this chapter, you will have a working environment which you can further configure and customize.

Requirements.

Uploading the Infrastructure Automation appliance file to Red Hat Virtualization requires:

- 44 GB of storage space on both the export domain and the local partition where **`/tmp`** resides, as the **`OVF`** archive is locally expanded into that directory.
- 12 GB RAM.
- 4 vCPUs.

Note:

See Creating a Virtual Machine in the Red Hat Virtualization *Virtual Machine Management Guide* for information on specifying memory for a new virtual machine.

Obtaining the Appliance

After the image completes downloading, you are ready to upload the Infrastructure Automation appliance to your Red Hat Virtualization environment and create a virtual machine.

Uploading the Appliance to Red Hat Virtualization

Upload the **`qcow2`** appliance image to your Red Hat Virtualization data storage domain from the Red Hat Virtualization Administration Portal.

- You must configure the Image I/O Proxy when running **`engine-setup`**. See Configuring the Red Hat Virtualization Manager in the *Red Hat Virtualization Installation Guide* for more information.
- Internet Explorer 10, Firefox 35, or Chrome 13 or greater is required to perform this upload procedure. Previous browser versions do not support the required HTML5 APIs.
- You must import the required certificate authority into the web browser used to access the Administration Portal.

Note:

To import the certificate authority, browse to **`https://<engine_address>/ovirt-engine/services/pki-resource?resource=ca-certificate&format=X509-PEM-CA`** and select all the trust settings. Refer to the instructions to install the certificate authority in Firefox, Internet Explorer, or Google Chrome.

To upload the appliance:

1. Log in to the Red Hat Virtualization Administration Portal.
2. Click **Storage > Disks**.
3. Select **Upload > Start**.
4. Click **Choose File**, and select the appliance image to upload.
5. Under **Disk Options**, enter an **Alias** for the image.
6. (Optional) Edit other fields in **Disk Options** if desired.
7. Click **OK**.

A progress bar in the **Status** column indicates the status of the upload. You can also pause, cancel, or resume uploads from the **Upload** menu.

The status shows **OK** when the image has completed uploading.

Note:

See the Uploading Images to a Data Storage Domain in the *Red Hat Virtualization Administration Guide* for more information.

1. If the upload times out and you see the message, `Reason: timeout due to transfer inactivity`, increase the timeout value:

   ```
   # engine-config -s TransferImageClientInactivityTimeoutInSeconds=6000
   ```

2. Restart the ovirt-engine service:

   ```
   # systemctl restart ovirt-engine
   ```

Creating a Infrastructure Automation Virtual Machine

After uploading the appliance to the data domain, create a virtual machine containing the Infrastructure Automation image and a second disk for the database:

1. In **Compute > Virtual Machines**, click **New** to open the **New Virtual Machine** dialog.
2. From the **General** tab, specify a name for the virtual machine and any other details.
3. Click **Attach**.
4. Select the Infrastructure Automation appliance you uploaded from the list of images and click **OK**.
5. Click the VM to open its details screen, and click the **Disks** tab.
6. Click **Edit** to the disk, and check **Bootable** if it is not marked already.
7. Click **OK**.
8. Add a second disk for the database:
   1. Click the **General** tab and click the **Edit** button to edit the virtual machine.
   2. Click **+** and **Create** to create a disk for the database (VMDB).
   3. Specify a **Size** in GB for the disk that allows sufficient space for your database. Red Hat recommends allocating the virtual machine disk fully at the time of creation. Three main factors affect the size of your database over time:
      - Virtual Machine Count: the most important factor in the calculation of virtual machine database (VMDB) size over time.
      - Host Count: the number of hosts associated with the provider.
      - Storage Count: the number of individual storage elements as seen from the perspective of the provider or host. It is not the total number of virtual disks for all virtual machines.
   4. Set the **Allocation Policy** to `Preallocated` (thick provisioning) for best performance.
   5. Specify any other values as desired.
9. Click **OK** to create the disk and return to the **New Virtual Machine** window.
10. Add a network interface to the virtual machine by selecting a vNIC profile from the **nic1** list.
11. Click **OK** to save your changes to the virtual machine.
12. Start the Infrastructure Automation appliance by clicking the **Run** button.

Your Red Hat Virtualization environment now contains a running Infrastructure Automation appliance.

Next, configure the database on the disk you created by following the steps in Configuring an Internal Database.

Configuring Infrastructure Automation

After installing Infrastructure Automation and running it for the first time, you must perform some basic configuration. You must complete these steps:

1. Add a disk to the infrastructure that is hosting your appliance.
2. Configure the database.
3. Configure messaging

Configure the appliance by using the internal appliance console.

Accessing the Appliance Console

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The Infrastructure Automation appliance summary screen displays.
3. Press **`Enter`** to manually configure settings.
4. Press the number for the item you want to change, and press **`Enter`**. The options for your selection are displayed.
5. Follow the prompts to make the changes.
6. Press **`Enter`** to accept a setting where applicable.

Note:

The Infrastructure Automation appliance console automatically logs out after five minutes of inactivity.

Configuring a Database

Infrastructure Automation uses a database to store information about the environment. Before using Infrastructure Automation, configure the database options for it; Infrastructure Automation provides the following two options for database configuration:

Configuring an Internal Database

Before installing an internal database, add a disk to the infrastructure hosting your appliance. See the documentation specific to your infrastructure for instructions for adding a disk. As a storage disk usually cannot be added while a virtual machine is running, Red Hat recommends adding the disk before starting the appliance. Infrastructure Automation only supports installing of an internal VMDB on blank disks; installation will fail if the disks are not blank.

1. Start the appliance and open a terminal console.
2. Enter the **`appliance_console`** command. The Infrastructure Automation appliance summary screen displays.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. You are prompted to create or fetch an encryption key.
    - If this is the first Infrastructure Automation appliance, choose **Create key**.
    - If this is not the first Infrastructure Automation appliance, choose **Fetch key from remote machine** to fetch the key from the first appliance. For worker and multi-region setups, use this option to copy key from another appliance.

      Note:

      All Infrastructure Automation appliances in a multi-region deployment must use the same key.
6. Choose **Create Internal Database** for the database location.

   In the **Configure Messaging** menu, select **Make No messaging changes**. If you see **`Configuration failed: Internal database require a volume mounted at /var/lib/pgsql. Please add an unpartitioned disk and try again.`** message, then ensure to add a second disk for the database per instructions as defined above.

7. Choose a disk for the database. This can be either a disk you attached previously, or a partition on the current disk.

   Red Hat recommends using a separate disk for the database.
   If there is an unpartitioned disk attached to the virtual machine, the dialog will show options similar to the following:

   ```
   1) /dev/vdb: 20480
   2) Don't partition the disk
   ```

   - Enter **1** to choose **/dev/vdb** for the database location. This option creates a logical volume using this device and mounts the volume to the appliance in a location appropriate for storing the database. The default location is **/var/lib/pgsql**, which can be found in the environment variable **$APPLIANCE_PG_MOUNT_POINT**.
   - Enter **2** to continue without partitioning the disk. A second prompt will confirm this choice. Selecting this option results in using the root filesystem for the data directory (not advised in most cases).

8. Enter **Y** or **N** for **Should this appliance run as a standalone database server?**

   - Select **Y** to configure the appliance as a database-only appliance. As a result, the appliance is configured as a basic PostgreSQL server, without a user interface.
   - Select **N** to configure the appliance with the full administrative user interface.

9. When prompted, enter a unique number to create a new region.

   Creating a new region destroys any existing data on the chosen database.
10. Create and confirm a password for the database.

Infrastructure Automation then configures the internal database. This takes a few minutes. After the database is created and initialized, you can log in to Infrastructure Automation.

Configuring an External Database

Based on your setup, you will choose to configure the appliance to use an external PostgreSQL database. For example, we can only have one database in a single region. However, a region can be segmented into multiple zones, such as database zone, user interface zone, and reporting zone, where each zone provides a specific function. The appliances in these zones must be configured to use an external database.

The **postgresql.conf** file requires specific settings for correct operation. For example, it must correctly reclaim table space, control session timeouts, and format the PostgreSQL server log for improved system support. It is recommended that external databases use a **postgresql.conf** file based on the standard file used by the Infrastructure Automation appliance.

Ensure you configure the settings in the **postgresql.conf** to suit your system. For example, customize the **shared_buffers** setting according to the amount of real storage available in the external system hosting the PostgreSQL instance. In addition, depending on the aggregate number of appliances expected to connect to the PostgreSQL instance, it may be necessary to alter the **max_connections** setting.

Note:

- Infrastructure Automation requires PostgreSQL version 13.
- **postgresql.conf** controls the operation of all databases managed by the PostgreSQL instance, therefore it is not recommended to run other databases on this PostgreSQL instance.

1. Start the appliance and open a terminal console.
2. Enter the **appliance_console** command. The appliance console summary screen will be displayed.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. Choose **Create Region in External Database** for the database location.
6. Enter the database hostname or IP address when prompted.
7. Enter the database name or leave blank for the default (**vmdb_production**).

8. Enter the database username or leave blank for the default (**root**).
9. Enter the chosen database user's password.
10. Confirm the configuration if prompted.

Infrastructure Automation will then configure the external database.

Configure Messaging

Configuring messaging is required for appliance setup. It is recommended to configure the broker on the same appliance where your database is configured

Note: You can only have one kafka broker per region

1. You can either configure the current appliance as a kafka broker, or point the appliance to an existing external kafka broker.

   Select the appropriate option either **Configure this appliance as a messaging server** or **Connect to an external messaging system** to connect to an external kafka broker. You will be asked to fill in the required **Message Client Parameters** like IP address and username/password.

2. Select **Proceed** and appliance_console will apply the configuration that you have requested then restart evmserverd to pick up the changes.

Configuring a Worker Appliance

You can use multiple appliances to facilitate horizontal scaling, as well as for dividing up work by roles. Accordingly, configure an appliance to handle work for one or many roles, with workers within the appliance carrying out the duties for which they are configured. You can configure a worker appliance through the terminal. The following steps demonstrate how to join a worker appliance to an appliance that already has a region configured with a database and messaging.

1. Start the appliance and open a terminal console.
2. Enter the **appliance_console** command. The Infrastructure Automation appliance summary screen displays.
3. Press **Enter** to manually configure settings.
4. Select **Configure Application** from the menu.
5. You are prompted to create or fetch a security key. Since this is not the first Infrastructure Automation appliance, choose **2) Fetch key from remote machine**. For worker and multi-region setups, use this option to copy the security key from another appliance.

   Note:

   All Infrastructure Automation appliances in a multi-region deployment must use the same key.

6. Choose **Join Region in External Database** for the database location.
7. Enter the database hostname or IP address when prompted.
8. Enter the port number or leave blank for the default (**5432**).
9. Enter the database name or leave blank for the default (**vmdb_production**).
10. Enter the database username or leave blank for the default (**root**).
11. Enter the chosen database user's password.
12. Confirm the configuration if prompted.
13. Choose **Connect to an external messaging system** to connect to the external kafka broker located on the appliance with the external database

    Note: You can only have one kafka broker per region

14. Enter the necessary **Message Client Parameters** such as the hostname/IP and username/password
15. Confirm the configuration if prompted.

Logging In After Installing Infrastructure Automation

Once Infrastructure Automation is installed, you can log in and perform administration tasks.

Log in to Infrastructure Automation for the first time after installing by:

1. Browse to the URL for the login screen. (https://xx.xx.xx.xx on the virtual machine instance)
2. Enter the default credentials (Username: **admin** | Password: **smartvm**) for the initial login.
3. Click **Login**.

Changing the Default Login Password

Change your password to ensure more private and secure access to Infrastructure Automation.

1. Browse to the URL for the login screen. (https://xx.xx.xx.xx on the virtual machine instance)
2. Click **Update Password** beneath the **Username** and **Password** text fields.
3. Enter your current **Username** and **Password** in the text fields.
4. Input a new password in the **New Password** field.
5. Repeat your new password in the **Verify Password** field.
6. Click **Login**.

Appendix

Appliance Console Command-Line Interface (CLI)

Currently, the **`appliance_console_cli`** feature is a subset of the full functionality of the **`appliance_console`** itself, and covers functions most likely to be scripted by using the command-line interface (CLI).

1. After starting the Infrastructure Automation appliance, log in with a user name of **`root`** and the default password of **`smartvm`**. This displays the Bash prompt for the root user.
2. Enter the **`appliance_console_cli`** or **`appliance_console_cli --help`** command to see a list of options available with the command, or simply enter **`appliance_console_cli --option <argument>`** directly to use a specific option.

Database Configuration Options

Option	Description
–region (-r)	region number (create a new region in the database - requires database credentials passed)
–internal (-i)	internal database (create a database on the current appliance)
–dbdisk	database disk device path (for configuring an internal database)
–hostname (-h)	database hostname
–port	database port (defaults to **`5432`**)
–username (-U)	database username (defaults to **`root`**)
–password (-p)	database password

–dbname (-d)	database name (defaults to **vmdb_production**)

v2_key Options

Option	Description
–key (-k)	create a new v2_key
–fetch-key (-K)	fetch the v2_key from the given host
–force-key (-f)	create or fetch the key even if one exists
–sshlogin	ssh username for fetching the v2_key (defaults to **root**)
–sshpassword	ssh password for fetching the v2_key

IPA Server Options

Option	Description
–host (-H)	set the appliance hostname to the given name
–ipaserver (-e)	IPA server FQDN
–ipaprincipal (-n)	IPA server principal (default: **admin**)
–ipapassword (-w)	IPA server password
–ipadomain (-o)	IPA server domain (optional). Will be based on the appliance domain name if not specified.
–iparealm (-l)	IPA server realm (optional). Will be based on the domain name of the ipaserver if not specified.
–uninstall-ipa (-u)	uninstall IPA client

Note:

- In order to configure authentication through an IPA server, in addition to using **Configure External Authentication (httpd)** in the **appliance_console**, external authentication can be optionally configured via the **appliance_console_cli** (command-line interface).
- Specifying **–host** will update the hostname of the appliance. If this step was already performed via the **appliance_console** and the necessary updates that are made to **/etc/hosts** if DNS is not properly configured, the **–host** option can be omitted.

Certificate Options

Option	Description
–ca (-c)	CA name used for certmonger (default: **ipa**)
–postgres-client-cert (-g)	install certs for postgres client
–postgres-server-cert	install certs for postgres server
–http-cert	install certs for http server (to create certs/httpd* values for a unique key)
–extauth-opts (-x)	external authentication options

Note: The certificate options augment the functionality of the **certmonger** tool and enable creating a certificate signing request (CSR), and specifying **certmonger** the directories to store the keys.

Other Options

Option	Description
–logdisk (-l)	log disk path
–tmpdisk	initialize the given device for temp storage (volume mounted at **/var/www/miq_tmp**)
–verbose (-v)	print more debugging info

Example Usage.

```
$ ssh root@appliance.test.company.com
```

To create a new database locally on the server by using **/dev/sdb**:

```
# appliance_console_cli --internal --dbdisk /dev/sdb --region 0 --password smartvm
```

To copy the v2_key from a host *some.example.com* to local machine:

```
# appliance_console_cli --fetch-key some.example.com --sshlogin root --sshpassword smartvm
```

You could combine the two to join a region where *db.example.com* is the appliance hosting the database:

```
# appliance_console_cli --fetch-key db.example.com --sshlogin root --sshpassword smartvm --
hostname db.example.com --password mydatabasepassword
```

To configure external authentication:

```
# appliance_console_cli --host appliance.test.company.com
                        --ipaserver ipaserver.test.company.com
                        --ipadomain test.company.com
                        --iparealm TEST.COMPANY.COM
                        --ipaprincipal admin
                        --ipapassword smartvm1
```

To uninstall external authentication:

```
# appliance_console_cli  --uninstall-ipa
```

Installing Infrastructure Automation for use with IBM Cloud Pak for AIOps

Follow these instructions for installing Infrastructure Automation for use with IBM Cloud Pak for AIOps.

- Offline Installation
  - Bastion host
  - Portable compute device
  - Portable storage device
- Online Installation
  - Post Installation

Online install of Infrastructure Automation for IBM Cloud Pak for AIOps

Follow these steps to complete an online install of Infrastructure Automation for use with IBM Cloud Pak for AIOps.

Infrastructure automation is included with IBM Cloud Pak® for AIOps. If you have a license for IBM Cloud Pak for AIOps, you are entitled to install and use Infrastructure automation. You can do the online installation of Infrastructure Automation once you have installed IBM Cloud Pak for AIOps.

This section explains the different steps and procedures you need to follow for the online installation of Infrastructure Automation for IBM Cloud Pak for AIOps.

Before you begin

Confirm your environment meets the requirements for Infrastructure Automation and your chosen installation method:

- Review the Planning section.

  Important: The **storageClass** and **storageClassLargeBlock** that are used for creating the IBM Cloud Pak for AIOps custom resource must have the same value as the **storageClass** and **storageClassLargeBlock** that are

used for creating the Infrastructure Automation custom resource.

- You need to make sure that your system has the minimum requirements needed for installing IBM Cloud Pak for AIOps or installing Infrastructure Automation. For instance, your architecture must be AMD64.

Note: The display names of some OpenShift console components, such as window titles and push buttons, vary between OpenShift versions. The following instructions are based on OpenShift version 4.8 console components.

Installation steps

From a high level, an installation of Infrastructure Automation consists of five steps:

1. Install IBM Cloud Pak for AIOps
2. Install the Infrastructure Automation operator
3. Create the custom resource for Infrastructure Automation
4. Verify the install
5. Log in to the IBM Cloud Pak Automation console
6. Deploying Infrastructure management
7. Assign user roles and permissions

1. Install IBM Cloud Pak for AIOps

You are recommended to have the IBM Cloud Pak for AIOps before you install Infrastructure Automation. If you have Infrastructure Automation installed in your cluster, you can install IBM Cloud Pak for AIOps within the same cluster. The following steps install IBM Cloud Pak for AIOps first.

Important: IBM Cloud Pak for AIOps and Infrastructure Automation must be installed in the same namespace. Installing IBM Cloud Pak for AIOps and Infrastructure Automation in separate namespaces is not supported.

You can install the IBM Cloud Pak for AIOps by choosing to use either the command line interface (CLI) or console. If your cluster is connected to the internet, then you can install IBM Cloud Pak for AIOps using the Red Hat® OpenShift® Container Platform console or the command line interface (CLI). Choose one of the following topics.

- Online production installation (console method)
- Online production installation (CLI method)

2. Install the Infrastructure Automation operator

For more information about operators, see Adding Operators to a cluster in the Red Hat OpenShift documentation.

1. Log in to your OpenShift cluster's console.
2. Click **Operators > OperatorHub.** The **OperatorHub** page is displayed.
3. In the **All Items** field, enter `IBM Infrastructure Automation.` The Infrastructure Automation operator is displayed.
4. Click the **IBM Infrastructure Automation** tile. The **IBM Infrastructure Automation** window is displayed.
5. Click **Install**. The **Install Operator page** is displayed.
6. Enter the following values:
   - Set the **Namespace** to be the project (namespace) in which to install the Operator, such as `cp4aiops`.
   - Set **Update Channel** to v4.2
   - Set **Approval Strategy** to *Automatic.*
7. Click **Install** and wait for the IBM Infrastructure Automation operator to install.
8. Verify that the IBM Infrastructure Automation is successfully installed.

   Navigate to **Operators > Installed Operators**, and select your project from the **Projects** dropdown. IBM Infrastructure Automation and its dependent operators in the project are listed with a status of *Succeeded.*

Important: IBM Cloud Pak for AIOps and Infrastructure Automation must be installed in the same namespace. Installing IBM Cloud Pak for AIOps and Infrastructure Automation in separate namespaces is not supported.

3. Create the custom resource for Infrastructure Automation

You can either create Infrastructure Automation custom resource with the default set of values that are provided by default or customize these value before you create Infrastructure Automation custom resource.

You would typically change the default value of Infrastructure Automation custom resource, if you plan to change one or more of the following for Managed services during installation of Managed services:

- Default pre-created secrets in Managed services
- DefaultManaged services proxy setting
- Default Managed services default github.com token to import templates
- Default HTTPS certificate for Managed services Management Console
- Configuring Managed services to set the SSH keys and known hosts in provider terraform pod
- Modify the default replica count of the pods in Managed services

1. Create Infrastructure Automation custom resource with customization

The following command creates an instance of the Infrastructure Automation custom resource called **`IAConfig`** with customization to Managed services. You must have your customization parameters under **`spec.manageservice`** section. The list of install parameters that can be customized along with its default values are listed in Managed services installation parameters page.

```
cat << EOF | oc apply -f -
kind: IAConfig
apiVersion: aiops.ibm.com/v1alpha1
metadata:
  name: ibm-ia-installer
  namespace: cp4aiops
spec:
  imagePullSecret: ibm-entitlement-key
  infraAutoComposableComponents:
  - enabled: <Set to true to install Infrastructure Management component of Infrastructure
Automation, false otherwise>
    name: ibm-management-im-install
    spec: {}
  - enabled: <Set to true to install Managed services component of Infrastructure Automation,
false otherwise>
    name: ibm-management-cam-install
    spec:
      manageservice:
        <Set your custom installation parameter values>
  license:
    accept: <Set true to accept the license>
  storageClass: <Storage Class name that supports RWX>
  storageClassLargeBlock: <Select a storage class with a large block size (for example, 64k)>
EOF
```

2. Create Infrastructure Automation custom resource with default values

Run the following command to create an instance of the Infrastructure Automation custom resource called **`IAConfig`**. The list of install parameters with its default values are listed in Managed services installation parameters page.

```
cat << EOF | oc apply -f -
kind: IAConfig
apiVersion: aiops.ibm.com/v1alpha1
metadata:
  name: ibm-ia-installer
  namespace: cp4aiops
spec:
  imagePullSecret: ibm-entitlement-key
  infraAutoComposableComponents:
  - enabled: <set to true to install Infrastructure Management component of Infrastructure
Automation, false otherwise>
    name: ibm-management-im-install
    spec: {}
  - enabled: <set to true to install Managed services component of Infrastructure Automation,
false otherwise>
    name: ibm-management-cam-install
    spec: {}
```

```
  license:
    accept: <set true to accept the license>
  storageClass: <Storage Class name that supports RWX>
  storageClassLargeBlock: <Select a storage class with a large block size (for example, 64k)>
EOF
```

After a few minutes, verify that the Infrastructure Automation operator is installed in the **cp4aiops** project (namespace) with the following command:

```
oc get pods -n cp4aiops | grep ibm-infrastructure-automation-operator-controller-manager
```

4. Verify the install

After a few minutes, run the following command to verify that your deployment is successful.

```
oc get iaconfigs.aiops.ibm.com -A
```

Example output:

```
NAME                PHASE         VERSION   STORAGECLASS   STORAGECLASSLARGEBLOCK   AGE
ibm-ia-installer    Running       4.2.0     rook-cephfs    rook-ceph-block          8m44s
```

5. Log in to the IBM Cloud Pak Automation console

1. Find the password for the *admin* username by running the following command:

   ```
   oc -n ibm-common-services get secret platform-auth-idp-credentials -o
   jsonpath='{.data.admin_password}' | base64 -d
   ```

2. Find the URL to access the IBM Cloud Pak Automation console with the following command.

   ```
   oc get route -n cp4aiops cpd -o jsonpath='{.spec.host}'
   ```

   The following output is a sample output:

   ```
   cpd-cp4aiops.apps.mycluster.mydomain
   ```

   Based on the sample output, your console URL would be **https://cpd-cp4aiops.apps.mycluster.mydomain**

6. Deploying Infrastructure management

You can deploy Infrastructure management in two ways, as a:

1. Containerized deployment
2. Virtual Machine

Deploying Infrastructure management as a containerized deployment (podified).

Complete these steps to install Infrastructure management as a containerized deployment.

- Prerequisites
- Deploy the ibm-management-im-install operand

Prerequisites

- Ensure the following operators for **Infrastructure management** have been installed.
  - **ibm-management-im-install** for Infrastructure management has been installed. For more information, see Infrastructure management.
- Ensure IBM Cloud Pak® is setup for LDAP authentication. You need an existing user-group from this LDAP repository to use for the Infrastructure management deployment.

Deploy the ibm-management-im-install operand

Install an instance of the Infrastructure management using the Red Hat® OpenShift® Container Platform console or the CLI (command-line tools).

1. Create an installation instance CR file **<im-install-cr.yaml>** using the following yaml file and update the **applicationDomain**, **imagePullSecret**, **initialAdminGroupName** and accept license.

```
apiVersion: infra.management.ibm.com/v1alpha1
kind: IMInstall
metadata:
  labels:
    app.kubernetes.io/instance: ibm-infra-management-install-operator
    app.kubernetes.io/managed-by: ibm-infra-management-install-operator
    app.kubernetes.io/name: ibm-infra-management-install-operator
  name: im-iminstall
  namespace: cp4aiops
spec:
  applicationDomain: <YOUR_IM_HTTPD_ROUTE>
  imagePullPolicy: Always
  imagePullSecret: ibm-entitlement-key
  initialAdminGroupName: <YOUR_LDAP_USER_GROUP>
  storageClassName: <STORAGE_CLASS_NAME>
  license:
    accept: true
```

   Where:

   YOUR_IM_HTTPD_ROUTE is a user-defined route, which must include a name for your installation plus part of your IBM Cloud Pak Automation console console's route. Use the following oc command to obtain the console route and modify it to derive your Infrastructure management route.

   ```
   oc get ingress.config.openshift.io -o=jsonpath='{.items[0].spec.domain}'
   ```

   Example output:

   ```
   apps.mycluster.myibm.com
   ```

   Add inframgmtinstall to the output to create YOUR_IM_HTTPD_ROUTE. For example: **inframgmtinstall.apps.mycluster.myibm.com**.

   YOUR_LDAP_USER_GROUP is an existing user-group defined in your LDAP repository. As part of the initial setup, this LDAP group is created in Infrastructure management to match your existing LDAP group by name, and assigned an account role which facilitates SSO login.

   Important: You must specify an LDAP user-group and it must contain at least one user that is able to login to the IBM Cloud Pak Automation console. For example, you have an existing LDAP group that is named group100 and a user with the username user100 is a member of the group. You enter group100 for the value of **<YOUR_LDAP_USER_GROUP>**.

   STORAGE_CLASS_NAME The storage class that will be used by the Infrastructure management data stores. You can specify the same storage class name you used for Infrastructure Automation's **storageClass**. If you do not specify the **storageClassName** keyword, then the cluster's default storage class will be used.

   ACCEPT LICENSE set the accept value to **true** to accept the license.

2. Deploy the installation instance CR yaml using one of the following methods:

   - Using the CLI login to your OpenShift cluster where Infrastructure Automation is installed. For example,

     ```
     oc login --token=`<sha256~EUe-BThLn-qRm32K16QDqLMCQZz5VKlM42JKIj2-U7M>` --
     server=https://`<api.mycluster.myibm.com>`:6443
     ```

     ```
     oc apply -f im-install-cr.yaml
     ```

   - Using the OpenShift Console.

     Create an IMInstall instance to deploy Infrastructure management.

     - Navigate to **Operators > Installed Operators.**
     - Switch the project to cp4aiops.
     - Click IBM Infrastructure Management Install.
     - Click Create Instance in the IMInstall tile.
     - Switch to YAML view.

- Copy and paste the installation instance yaml created in step 1.
- Click Create.

3. Verify the pods are running (It can take a few minutes for the pods to start) with the command,

   ```
   oc get pods -n cp4aiops
   ```

   The following Infrastructure management pods should be running.

   - 1-event-handle-*
   - 1-generic-*
   - 1-priority-*
   - 1-remote-console-*
   - 1-reporting-*
   - 1-schedule-*
   - 1-ui-*
   - 1-web-service-*
   - httpd-*
   - ibm-infra-management-application-*
   - memcached-*

Deploying Infrastructure management as a virtual machine appliance

Complete these steps to install Infrastructure management as a virtual machine appliance.

- Prerequisites
- Step 1. Download the appliance image
- Step 2. Install and configure the appliance
- Step 3. Configure OIDC for Infrastructure management appliance

Prerequisites

- Ensure you enable the operators for **`Infrastructure management`** by opening the installation YAML file. Locate the **`pakModules`** section, and change **`enabled: false`** to **`enabled: true`**. Enable these Infrastructure management-related operators:
  - **`ibm-management-im-install`** for Infrastructure management has been installed. For more information, see Infrastructure management.
- You must configure and connect an LDAP directory with IBM Cloud Pak®. You must have an LDAP group in your configuration for IBM Cloud Pak with users defined who will access Infrastructure management.

Step 1. Download the Infrastructure management appliance package for your environment.

Follow the steps in Install the Infrastructure management appliance for your virtual environment.

Step 2. Install and configure the Infrastructure management appliance.

Follow the steps in Install the Infrastructure management appliance for your virtual environment.

Step 3. Configure OIDC integration between IBM Cloud Pak® and the Infrastructure management appliance.

Note: Only OIDC-based authentication (OpenID Connect) is supported. The configuration of OpenID Connect (OIDC) is required for integration with Infrastructure management and IBM Cloud Pak®.

- Prerequisites
- Step 1. Register Infrastructure management instance with IAM as an OIDC client
- Step 2. Import the Root CA certificate to Infrastructure management
- Step 3. Configure the Administrative UI
- Step 4. Integrating Infrastructure management with IBM Cloud Pak for AIOps

Prerequisites

- Before you configure OIDC make sure you install the Infrastructure management virtual machine appliance. For more information, see Installation and Upgrade.
- Single sign-on with Infrastructure management and IBM Cloud Pak for AIOps requires an LDAP server connection.

  For more information about adding an LDAP connection, see Configuring LDAP connection.

Step 1. Register Infrastructure management instance with IAM as an OIDC client

In order to enable single sign-on (SSO) for Infrastructure management with OIDC, the Infrastructure management instance needs to register as an OIDC client with Identity and Access Management (IAM). Complete these steps on the IBM Cloud Pak for AIOps cluster.

You can register Infrastructure management as an OIDC client with IAM using the **`cloudctl`** command.

The registration method requires the following registration payload in a file "registration.json":

```
{
  "token_endpoint_auth_method":"client_secret_basic",
  "client_id": "<YOUR_CLIENT_ID>",
  "client_secret": "<YOUR_CLIENT_SECRET>",
  "scope":"openid profile email",
  "grant_types":[
     "authorization_code",
     "client_credentials",
     "password",
     "implicit",
     "refresh_token",
     "urn:ietf:params:oauth:grant-type:jwt-bearer"
  ],
  "response_types":[
     "code",
     "token",
     "id_token token"
  ],
  "application_type":"web",
  "subject_type":"public",
  "post_logout_redirect_uris":[
     "https://<YOUR_CLOUD_PAK_ROUTE>"   ],
  "preauthorized_scope":"openid profile email general",
  "introspect_tokens":true,
  "trusted_uri_prefixes":[
     "https://<YOUR_CLOUD_PAK_ROUTE>/"    ],
  "redirect_uris":
["https://<YOUR_CLOUD_PAK_ROUTE>/auth/liberty/callback","https://<INFRA_MGMT_URL>/oidc_login/r
edirect_uri"]
}
```

Example registration payload (for reference only):

```
{
  "token_endpoint_auth_method":"client_secret_basic",
  "client_id": "AaNzNVFsSjlLVkl6Zk5hZ01MRzJVaVdnbFcxNGl5cnQK",
  "client_secret": "AaNVNzF4ZUxNSVBQUHZHdG1xQmNsTTFOWmNUUGlnYUkK",
  "scope":"openid profile email",
  "grant_types":[
     "authorization_code",
     "client_credentials",
     "password",
     "implicit",
     "refresh_token",
     "urn:ietf:params:oauth:grant-type:jwt-bearer"
  ],
  "response_types":[
     "code",
     "token",
     "id_token token"
  ],
  "application_type":"web",
  "subject_type":"public",
  "post_logout_redirect_uris":["https://cp-console.apps.mycluster.mydomain.com"],
```

```
  "preauthorized_scope":"openid profile email general",
  "introspect_tokens":true,
  "trusted_uri_prefixes":["https://cp-console.apps.mycluster.mydomain.com/"],
  "redirect_uris":["https://cp-
console.apps.mycluster.mydomain.com/auth/liberty/callback","https://im-
mycluster.mydomain.com/oidc_login/redirect_uri"]
}
```

1. Create a file named **registration.json** from the example template. Replace the values in the example template payload registration with the actual values based on your installation.

   - **YOUR_CLIENT_ID** Your base64 encoded character string.
   - **YOUR_CLIENT_SECRET** Your base64 encoded character string.

     Note: The **<YOUR_CLIENT_ID>** and **<YOUR_CLIENT_SECRET>** need to be generated. The values can be any string, but normally a 32 character string that is base64 encoded is used. You can use BASE64 to encode your character string. For more information, see: BASE64. Make a note of the values you generate for **<YOUR_CLIENT_ID>** and **<YOUR_CLIENT_SECRET>**. You will use the values in the next section to update the Apache configuration file.

     Example command that uses base64 to encode a character string:

     ```
     #
     # Generate two encrypted streams from some longer-than-32-characters strings
     #
     echo There is a huge white elephant in LA zoo |base64
     echo 12345678901234567890123456789012345 |base64
     ```

   - **YOUR_CLOUD_PAK_ROUTE** The URL of the IBM Cloud Pak for AIOps console.
   - **INFRA_MGMT_URL** The URL of the Infrastructure management virtual machine appliance.
   - **post_logout_redirect_uris** The URL of the IBM Cloud Pak for AIOps console.
   - **trusted_uri_prefixes** The URL of the IBM Cloud Pak for AIOps console with "forward slash" /.
   - **redirect_uris** The URL of the IBM Cloud Pak for AIOps console with the path to call back and the URL of the Infrastructure management host with the path to the redirect_uri.

   Note: You can run the following command on the IBM Cloud Pak for AIOps cluster to determine the URL of the IBM Cloud Pak for AIOps console:

   ```
   oc get routes cp-console -o=jsonpath='{.spec.host}' -n ibm-common-services
   ```

2. After the file **registration.json** is completed, log in and run the command to register Infrastructure management as an OIDC client.

   Note: Include the **-n kube-system** to specify this project, or the **cloudctl iam** command can fail.

   ```
   cloudctl login -a https://<YOUR_CLOUD_PAK_ROUTE> -n kube-system
   ```

   Example **cloudctl iam** command:

   ```
   cloudctl iam oauth-client-register -f registration.json
   ```

Step 2. Import the Root CA certificate to the Infrastructure management appliance

1. Retrieve the cluster CA cert by running this command on the cluster:

   ```
   oc get secret -n ibm-common-services ibmcloud-cluster-ca-cert -o
   jsonpath='{.data.ca.crt}'| base64 --decode
   ```

   Note: When copying this command there can be some additional characters added that can cause incorrect command syntax. You can copy the first portion **oc get secret -n ibm-common-services ibmcloud-cluster-ca-cert -o jsonpath=** and manually add the remaining syntax **'{.data.ca\.crt}'| base64 --decode**.

2. Copy and paste the output to a file, for example **ibm_cp_im.crt**

3. Edit the **ibm_cp_im.crt** file, and change:

   - **BEGIN CERTIFICATE** to **BEGIN TRUSTED CERTIFICATE**
   - **END CERTIFICATE** to **END TRUSTED CERTIFICATE**

   Note: The following steps must be completed by logging in to the Infrastructure management appliance system as root user:

4. Copy the updated **ibm_cp_im.crt** file to the Infrastructure management appliance and save it in the directory: **/etc/pki/ca-trust/source/anchors**

5. Run the command:

   ```
   update-ca-trust
   ```

6. Restart the evm server by running the command:

   ```
   systemctl restart evmserverd
   ```

7. Copy the Apache OIDC template configuration file with these steps:

   ```
   export TEMPLATE_DIR="/opt/IBM/infrastructure-management-appliance/TEMPLATE"

   cp ${TEMPLATE_DIR}/etc/httpd/conf.d/manageiq-remote-user-openidc.conf /etc/httpd/conf.d/

   cp ${TEMPLATE_DIR}/etc/httpd/conf.d/manageiq-external-auth-openidc.conf.erb
   /etc/httpd/conf.d/manageiq-external-auth-openidc.conf
   ```

8. The Apache **/etc/httpd/conf.d/manageiq-external-auth-openidc.conf** configuration file must be updated with installation-specific values. Replace the contents of the file with the actual values based on the installation.

   Sample configuration file:

   ```
   LoadModule           auth_openidc_module modules/mod_auth_openidc.so
   ServerName           https://<YOUR_IM_APPLIANCE_HOSTNAME>
   LogLevel             warn

   OIDCClientID                    <YOUR_CLIENT_ID>
   OIDCClientSecret                <YOUR_CLIENT_SECRET>
   OIDCRedirectURI
   https://<YOUR_IM_APPLIANCE_HOSTNAME>/oidc_login/redirect_uri
   OIDCCryptoPassphrase            <PASSPHRASE>
   OIDCOAuthRemoteUserClaim        sub
   OIDCRemoteUserClaim             name

   OIDCOAuthClientID                   <YOUR_CLIENT_ID>
   OIDCOAuthClientSecret               <YOUR_CLIENT_SECRET>
   OIDCOAuthIntrospectionEndpoint
   https://<YOUR_CLOUD_PAK_ROUTE>/idprovider/v1/auth/introspect
   OIDCOAuthIntrospectionEndpointAuth client_secret_basic

   OIDCProviderIssuer                  https://127.0.0.1:443/idauth/oidc/endpoint/OP
   OIDCProviderAuthorizationEndpoint
   https://<YOUR_CLOUD_PAK_ROUTE>/idprovider/v1/auth/authorize
   OIDCProviderTokenEndpoint
   https://<YOUR_CLOUD_PAK_ROUTE>/idprovider/v1/auth/token
   OIDCProviderJwksUri                 https://<YOUR_CLOUD_PAK_ROUTE>/idprovider/v1/auth/jwk
   OIDCProviderEndSessionEndpoint
   https://<YOUR_CLOUD_PAK_ROUTE>/idprovider/v1/auth/logout

   OIDCScope                         "openid email profile"
   OIDCResponseMode                  "query"
   OIDCProviderTokenEndpointAuth      client_secret_post

   OIDCPassUserInfoAs json
   OIDCSSLValidateServer off
   OIDCOAuthSSLValidateServer off
   OIDCHTTPTimeoutShort 10

   OIDCCacheEncrypt On
   <Location /oidc_login>
     AuthType  openid-connect
   ```

```
  Require   valid-user
</Location>

<Location /ui/service/oidc_login>
  AuthType openid-connect
  Require valid-user
  Header set Set-Cookie "miq_oidc_access_token=%{OIDC_access_token}e; Max-Age=10;
Path=/ui/service"
</Location>

<LocationMatch ^/api(?!\/(v[\d\.]+\/)?product_info$)>
  SetEnvIf X-Auth-Token  '^.+$'                let_api_token_in
  SetEnvIf X-MIQ-Token   '^.+$'                let_sys_token_in
  SetEnvIf X-CSRF-Token  '^.+$'                let_csrf_token_in

  AuthType  oauth20
  AuthName  "External Authentication (oidc) for API"

  Require   valid-user
  Order     Allow,Deny
  Allow from env=let_api_token_in
  Allow from env=let_sys_token_in
  Allow from env=let_csrf_token_in
  Satisfy   Any
</LocationMatch>

RequestHeader unset X-REMOTE-USER
RequestHeader unset X-REMOTE_USER
RequestHeader unset X_REMOTE-USER
RequestHeader unset X_REMOTE_USER
RequestHeader set X_REMOTE_USER %{OIDC_CLAIM_PREFERRED_USERNAME}e
env=OIDC_CLAIM_PREFERRED_USERNAME
RequestHeader set X_EXTERNAL_AUTH_ERROR %{EXTERNAL_AUTH_ERROR}e env=EXTERNAL_AUTH_ERROR
RequestHeader set X_REMOTE_USER_EMAIL %{OIDC_CLAIM_EMAIL}e env=OIDC_CLAIM_EMAIL
RequestHeader set X_REMOTE_USER_FIRSTNAME %{OIDC_CLAIM_GIVEN_NAME}e
env=OIDC_CLAIM_GIVEN_NAME
RequestHeader set X_REMOTE_USER_LASTNAME %{OIDC_CLAIM_FAMILY_NAME}e
env=OIDC_CLAIM_FAMILY_NAME
RequestHeader set X_REMOTE_USER_FULLNAME %{OIDC_CLAIM_NAME}e env=OIDC_CLAIM_NAME
RequestHeader set X_REMOTE_USER_GROUPS %{OIDC_CLAIM_GROUPS}e env=OIDC_CLAIM_GROUPS
RequestHeader set X_REMOTE_USER_DOMAIN %{OIDC_CLAIM_DOMAIN}e env=OIDC_CLAIM_DOMAIN
```

- `YOUR_IM_APPLIANCE_HOSTNAME` Specifies the hostname of the Infrastructure management appliance server.
- `YOUR_CLIENT_ID` The client ID used for registering Infrastructure management as an OIDC client with IAM.
- `YOUR_CLIENT_SECRET` The client secret that is used for registering Infrastructure management as an OIDC client with IAM.
- `YOUR_CLOUD_PAK_ROUTE` The URL of the IBM Cloud Pak UI console.
- `OIDCCryptoPassphrase` Can be any arbitrary alpha-numeric string.

Note: `YOUR_CLIENT_ID` and `YOUR_CLIENT_SECRET` values are generated when you register Infrastructure Management as an OIDC client.

Example configuration file (for reference only):

```
LoadModule          auth_openidc_module modules/mod_auth_openidc.so
ServerName          https://im-mycluster.mydomain.com
LogLevel            warn

OIDCCLientID                        N3NzNVFsSjlLVkl6Zk5hZ01MRzJVaVdnbFcxNGl5cnQK
OIDCClientSecret                    VWNVNzF4ZUxNSVBQUHZHdG1xQmNsTTFOWmNUUGlnYUkK
OIDCRedirectURI                     https://im-
mycluster.mydomain.com/oidc_login/redirect_uri
OIDCCryptoPassphrase                alphabeta
OIDCOAuthRemoteUserClaim            sub
OIDCRemoteUserClaim                 name

OIDCOAuthClientID                   N3NzNVFsSjlLVkl6Zk5hZ01MRzJVaVdnbFcxNGl5cnQK
OIDCOAuthClientSecret               VWNVNzF4ZUxNSVBQUHZHdG1xQmNsTTFOWmNUUGlnYUkK
OIDCOAuthIntrospectionEndpoint      https://cp-
console.apps.mycluster.mydomain.com/idprovider/v1/auth/introspect
OIDCOAuthIntrospectionEndpointAuth client_secret_basic
```

```
OIDCProviderIssuer                  https://127.0.0.1:443/idauth/oidc/endpoint/OP
OIDCProviderAuthorizationEndpoint  https://cp-
console.apps.mycluster.mydomain.com/idprovider/v1/auth/authorize
OIDCProviderTokenEndpoint          https://cp-
console.apps.mycluster.mydomain.com/idprovider/v1/auth/token
OIDCProviderJwksUri                https://cp-
console.apps.mycluster.mydomain.com/idprovider/v1/auth/jwk
OIDCProviderEndSessionEndpoint     https://cp-
console.apps.mycluster.mydomain.com/idprovider/v1/auth/logout

OIDCScope                          "openid email profile"
OIDCResponseMode                   "query"
OIDCProviderTokenEndpointAuth      client_secret_post

OIDCPassUserInfoAs json
OIDCSSLValidateServer off
OIDCOAuthSSLValidateServer off
OIDCHTTPTimeoutShort 10

OIDCCacheEncrypt On
<Location /oidc_login>
  AuthType  openid-connect
  Require   valid-user
</Location>

<Location /ui/service/oidc_login>
  AuthType openid-connect
  Require valid-user
  Header set Set-Cookie "miq_oidc_access_token=%{OIDC_access_token}e; Max-Age=10;
Path=/ui/service"
</Location>

<LocationMatch ^/api(?!\/(v[\d\.]+\/)?product_info$)>
  SetEnvIf X-Auth-Token  '^.+$'                let_api_token_in
  SetEnvIf X-MIQ-Token   '^.+$'                let_sys_token_in
  SetEnvIf X-CSRF-Token  '^.+$'                let_csrf_token_in

  AuthType  oauth20
  AuthName  "External Authentication (oidc) for API"

  Require   valid-user
  Order     Allow,Deny
  Allow from env=let_api_token_in
  Allow from env=let_sys_token_in
  Allow from env=let_csrf_token_in
  Satisfy   Any
</LocationMatch>

RequestHeader unset X-REMOTE-USER
RequestHeader unset X-REMOTE_USER
RequestHeader unset X_REMOTE-USER
RequestHeader unset X_REMOTE_USER
RequestHeader set X_REMOTE_USER %{OIDC_CLAIM_PREFERRED_USERNAME}e
env=OIDC_CLAIM_PREFERRED_USERNAME
RequestHeader set X_EXTERNAL_AUTH_ERROR %{EXTERNAL_AUTH_ERROR}e env=EXTERNAL_AUTH_ERROR
RequestHeader set X_REMOTE_USER_EMAIL %{OIDC_CLAIM_EMAIL}e env=OIDC_CLAIM_EMAIL
RequestHeader set X_REMOTE_USER_FIRSTNAME %{OIDC_CLAIM_GIVEN_NAME}e
env=OIDC_CLAIM_GIVEN_NAME
RequestHeader set X_REMOTE_USER_LASTNAME %{OIDC_CLAIM_FAMILY_NAME}e
env=OIDC_CLAIM_FAMILY_NAME
RequestHeader set X_REMOTE_USER_FULLNAME %{OIDC_CLAIM_NAME}e env=OIDC_CLAIM_NAME
RequestHeader set X_REMOTE_USER_GROUPS %{OIDC_CLAIM_GROUPS}e env=OIDC_CLAIM_GROUPS
RequestHeader set X_REMOTE_USER_DOMAIN %{OIDC_CLAIM_DOMAIN}e env=OIDC_CLAIM_DOMAIN
```

9. Restart Apache on the appliance.

```
systemctl restart httpd
```

Step 3. Configure the Administrative UI

Update the Appliance Administrative UI to be OIDC aware and function. Complete these steps on each UI-enabled Infrastructure management appliance.

1. Log in as **admin**.
2. Select the **Settings** > **Application Settings**, select **"Server "EVM [1]" (current)"** under Zones, then select the **Authentication** tab.
3. In the **Authentication** section, set the **Mode** to **External (httpd)**
4. In the **External Authentication (HTTPd) Settings** section, set **Provider Type** to **Enable OpenID-Connect**.
    - **Note:** This setting enables the OIDC login button on the login screen that redirects to the OIDC protected page for authentication, and supports the OIDC logout process.
5. Optional: In the **External Authentication (HTTPd) Settings** section, select **Enable Single Sign-On**.
    - **Note:** If you select this option, the initial access to the Appliance Administrative UI will redirect to the OIDC Identity Provider authentication screen.
6. In the **Role Settings** section, select the **Get User Groups from External Authentication (HTTPd)** setting.
7. Click Save.
8. Select **Access Control**, click **Groups** and make sure the user's groups are created on the Appliance and appropriate roles are assigned to those groups. The user's groups to be added in Infrastructure management should have the same names as the groups defined in the LDAP server that is configured in the IBM Cloud Pak console.
    1. Under the **Access Control**, click **Groups**.
    2. Click **Configuration**, then **Add a new Group**.
    3. Enter your existing LDAP group name in the **Description** field. For example, **im_ldap_group**.
    4. Select the **Role** for this LDAP group. For example, select the **EvmRole-super_administrator** to map to this group. The Infrastructure management administrator can map to the roles that make sense for the user group.
    5. Select **My Company** for **Project/Tenant**.
    6. Click Add.

    Note: Access control in Infrastructure management is based on group membership as roles are assigned to groups. When Infrastructure management is integrated with IBM Cloud Pak for AIOps using single sign-on (SSO), it looks at the user's group membership in the identity token and checks if that group exists in Infrastructure management. If the group doesn't exist, then access is denied. You must create at least one group with the role **EvmRole-super_administrator** and assign one user to this group. This user will be the Admin user for Infrastructure management and referred to as **IM_ADMIN_LDAP_USER**. The **IM_ADMIN_LDAP_USER** must be a member of the LDAP group with the role **EvmRole-super_administrator** in Infrastructure management.

    Important: You must create the groups in Infrastructure management that match your existing LDAP groups by name, and assign the groups account roles. At least one group to which the user belongs in LDAP that IBM Cloud Pak for AIOps is configured to use must also be created in Infrastructure management. You must assign a proper role to this group in Infrastructure management.

    Example: In LDAP a group that is named **imgroup100** exists and a user with username **imuser100** is a member of the group. The user **imuser100** and the group **imgroup100** must be created in Infrastructure management.

Step 4. Integrating Infrastructure management with IBM Cloud Pak for AIOps

Enable navigation to Infrastructure management within the IBM Cloud Pak console.

Complete the following steps on a Linux system. You can use the boot node from the HUB cluster where IBM Cloud Pak for AIOps is installed. These steps enable navigation to Infrastructure management from the IBM Cloud Pak console:

1. Clone the IBM Cloud Pak for AIOps samples GitHub repository to obtain the menu customization script. You must run the script on a Linux operating system.

    Download the supporting subdirectories to make the script run successfully. The files and subdirectories that are required to run the script are

```
im-appliance-link-install.sh
nav/automate.yml
nav_template/automate.sh
```

2. Install and authenticate **kubectl**. For more information, see Installing the Kubernetes CLI (kubectl).

3. Navigate to the directoy **cp4waiops-samples/infrastructure-automation/infrastructure-management/appliance**. Set the file permissions on the script and run **im-appliance-link-install.sh** to enable navigation to your Infrastructure management instance:

   ```
   chmod 755 ./im-appliance-link-install.sh
   ```

   ```
   ./im-appliance-link-install.sh -i <Infrastructure management URL>
   ```

   - **-i** Is a required parameter that refers to the URL for the Infrastructure management console. For example, **./im-appliance-link-install.sh -i https://im-mycluster.mydomain.com/**

4. Verify that the Infrastructure management instance is in the IBM Cloud Pak console navigation menu. From the IBM Cloud Pak navigation menu, click **Automate infrastructure** > **Infrastructure management**.

Infrastructure management is integrated with the IBM Cloud Pak console.

7. Assign user roles and permissions

When you install Infrastructure Automation and deploy Infrastructure management, you, or an administrator, must add the required Kubernetes permissions to user roles before your users can begin to access and use Infrastructure Automation tools, such as Managed services or the Service catalog. For instance, users that do not have an Administrator role are not able to use the Infrastructure management Managed services and Service Catalog or create user groups. For more information about how to add permissions to a role, see Managing roles for Infrastructure Automation.

Installing IBM Cloud Pak for AIOps Infrastructure Automation for use with IBM Cloud Pak for AIOps in an air-gapped environment (offline)

If your cluster is not connected to the internet, you can complete a production install of IBM Cloud Pak® for AIOps Infrastructure Automation in your cluster by using either a bastion host, portable compute device, or a portable storage device.

The IBM Catalog Management Plug-in for IBM Cloud Pak® is now used for the deployment of IBM Cloud Paks® in a disconnected environment. It simplifies the process for discovering IBM product images, uses standard tools for registry and cluster access, and extends the Red Hat OpenShift CLI (**oc**) capability to streamline the air-gapped deployment process.

Overview

It is common in production to have a cluster that does not have internet access. In these cases, you can still install Infrastructure Automation and Red Hat® OpenShift® Container Platform in an air-gapped (otherwise known as offline or disconnected) environment. Unlike online installations, air-gapped installations require you to enable the IBM Operator Catalog to mimic a typical online installation by using images in your own registry.

The following diagram provides an overview of air-gapped installation scenarios for portable compute, portable storage, and bastion hosts:

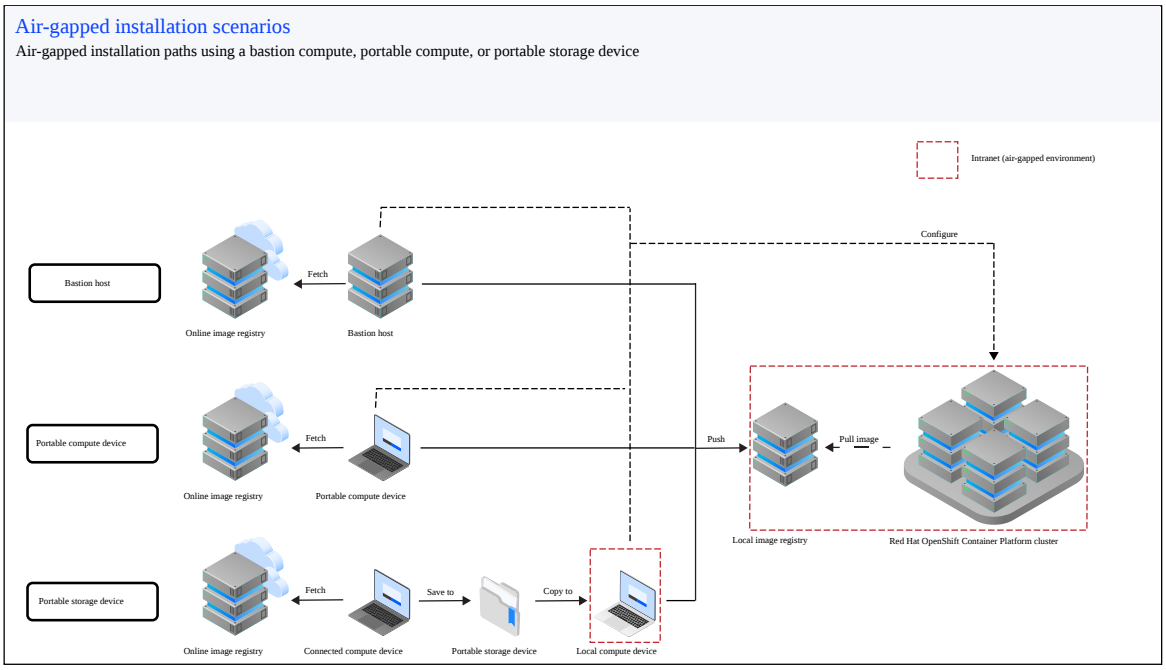


Figure. Air-gap scenario

All of these scenarios use Container Application Software for Enterprises (CASE) files to mirror content from a source to a target. CASE is a specification that defines metadata and structure for packaging, managing, and unpacking containerized applications.

You can store the product code and images to a portable compute device (like a laptop), portable storage device (like an external hard disk drive), or a bastion host and then transfer them to a local air-gapped network. If you are using either a portable compute device or a portable storage device, you don't need a bastion host to install in your air-gapped environment.

The images that are stored are large, and you must ensure that each of your airgap devices have enough storage, as follows.

- Bastion host method - the local registry and the bastion host must each have at least 58 GB of storage space.
- Portable compute device method - the local registry and the portable compute device must each have at least 58 GB of storage space.
- Portable storage device - the local registry, connected compute device, portable storage device, and the local compute device must each have at least 58 GB of storage space.

Table 1. Air-gapped installation scenarios

Air-gapped installation approach	Example	Description
Bastion host	Bastion host	A bastion server is a device that has access to both the public internet and the local intranet where a local registry and Red Hat OpenShift Container Platform clusters are located. Using the bastion server, you can replicate your images through the bastion server directly to the local, intranet registry behind the firewall.
Portable compute device	Laptop	A portable compute device, such as a laptop, can be used to download images from the entitled registry to a portable container registry running locally on the device. You can then bring the device behind your firewall and copy the images from your portable registry on the device to the local, intranet registry behind the firewall.
Portable storage device	Portable hard disk drive	Images are mirrored to a container registry on a connected compute device, and then copied to a portable storage device. The portable storage device is plugged into a local compute device inside your firewall, and the images are copied to the local compute device. The images from the local compute device are then mirrored into the local, intranet registry behind the firewall.

From a high level, an air-gapped installation of Infrastructure Automation consists of five steps:

1. Set up your image registry access and mirroring environment (One-time action)
2. Set environment variables and download CASE files

3. Mirror images depending on installation scenario
4. Configure storage
5. Install Infrastructure Automation by way of Red Hat OpenShift Container Platform

Installation methods

The following sections illustrate the required steps for each approach to an air-gapped installation. Pick the appropriate methodology and follow the steps within that section. Step ordering might vary depending on your methodology:

- Bastion host
- Portable compute device
- Portable storage device

Preparing to install IBM Cloud Pak for AIOps Infrastructure Automation in an air-gapped environment (offline) using a bastion host

If your cluster is not connected to the internet, you can deploy a production install of IBM Cloud Pak® for AIOps Infrastructure Automation on your cluster by using a bastion host.

In this scenario, your air-gapped (offline) environment has a target registry, and a Red Hat® OpenShift® Container Platform cluster on which Infrastructure Automation is to be installed. The bastion host has access to the internet and to the offline environment. Infrastructure Automation images are mirrored from the internet through the bastion server to the target registry in the offline environment. Infrastructure Automation can then be installed in the offline environment by using the target registry.

Notes: The following procedure is based on a Red Hat® OpenShift® Container Platform 4.12 environment and includes links for that version. If your environment uses a different supported version of Red Hat® OpenShift® Container Platform, ensure that you follow the Red Hat OpenShift documentation for that version.

Before you begin

Review the Planning section. Your environment must meet the system requirements.

Important: The **`storageClass`** and **`storageClassLargeBlock`** that are used for creating the IBM Cloud Pak for AIOps custom resource must have the same value as the **`storageClass`** and **`storageClassLargeBlock`** that are used for creating the Infrastructure Automation custom resource.

Procedure

From a high level, an air-gapped installation of Infrastructure Automation consists of five steps:

1. Set up your mirroring environment
2. Set environment variables and download CASE files
3. Mirror images to your final location
4. Configure storage
5. Install Infrastructure Automation by way of Red Hat OpenShift Container Platform

1. Set up your mirroring environment

Before you install any IBM Cloud Pak® on an air-gapped environment, you must set up a host that can be connected to the internet to complete configuring your mirroring environment. To set up your mirroring environment, complete the following steps:

Prerequisites

You must satisfy the following prerequisites:

- Red Hat® OpenShift® Container Platform requires you to have cluster admin access to run the **`configure-cluster-airgap`** and **`install-operator`** commands.
- Allow access to the following sites and ports:

Table 1. Sites and ports that must be accessible

Site	Description
• `icr.io` • `cp.icr.io` • `dd0.icr.io` • `dd2.icr.io` • `dd4.icr.io` • `dd6.icr.io`	Allow access to these hosts on port 443 to enable access to the **`IBM Cloud Container Registry`**, CASE OCI artifact, and IBM Cloud Pak® foundational services catalog source.
• `dd1-icr.ibm-zh.com` • `dd3-icr.ibm-zh.com` • `dd5-icr.ibm-zh.com` • `dd7-icr.ibm-zh.com`	If you are located in China, also allow access to these hosts on port 443.
`github.com`	Github houses CASE files, IBM Cloud Pak tools and scripts.
`redhat.com`	Red Hat OpenShift registries that are required for Configuring your firewall for OpenShift Container Platform and for Red Hat OpenShift upgrades.

1.1 Install and configure Red Hat OpenShift

Infrastructure Automation requires Red Hat OpenShift to be installed and running on your target cluster. You must have administrative access to your Red Hat OpenShift cluster.

For more information about the supported versions of Red Hat OpenShift, see Supported Red Hat OpenShift Container Platform versions.

1. Install Red Hat OpenShift by using the instructions in the Red Hat OpenShift documentation. Information about installing a cluster in a restricted network is given in Mirroring images for a disconnected installation..
2. Install the Red Hat OpenShift command line interface (**`oc`**) on your cluster's boot node and run **`oc login`**. For more information, see the instructions in Getting started with the Red Hat OpenShift CLI.
3. Optionally configure a custom certificate for Infrastructure Automation to use. You can use either of the following methods:
   - Configure a custom certificate for the Red Hat OpenShift cluster. Follow the instructions in the Red Hat OpenShift documentation Replacing the default ingress certificate. Then, deploy the signing CA certificate into the cluster by following the instructions in the Red Hat OpenShift documentation Replacing the CA Bundle certificate.
   - If you would like to use a custom certificate for Infrastructure Automation only, then after installation is complete follow the instructions in Using a custom certificate.

1.2 Set up a target registry

You must have a local Docker type production-grade registry available to store the Infrastructure Automation images in. The registry must meet the following requirements:

- supports Docker Manifest V2, Schema 2.
- supports multi-architecture images.
- is accessible from the Red Hat OpenShift cluster nodes.
- allows path separators in the image name.
- you have the username and password for a user who can read from and write to the registry.

If you do not already have a suitable production-grade registry available, then you must install and configure one. For more information, see About the mirror registry in the Red Hat OpenShift documentation.

Important: Do not use the Red Hat OpenShift image registry as your target registry. The Red Hat OpenShift registry does not support multi-architecture images or path separators in the image name.

1.3 Prepare a host

You must prepare a bastion host that can connect to the internet and to the air-gapped network with access to the Red Hat® OpenShift® Container Platform cluster and the local, intranet Docker type registry. Your host must be on a Linux® x86_64 or Mac platform with any operating system that the IBM Cloud Pak® CLI and the Red Hat® OpenShift® Container Platform CLI support. If you are on a Windows platform, you must run the actions in a Linux® x86_64 VM or from a Windows Subsystem for Linux (WSL) terminal.

Your host must have 58 GB storage to hold all of the software that is to be transferred to the local, intranet Docker type registry.

The following table explains the software requirements for installing Infrastructure Automation in an air-gapped environment:

Table 2. Software requirements and purpose

Software	Purpose
OpenSSL	Validating certificates when you run the air-gapped install scripts
Docker	Container management
Podman	Container management
Apache httpd-tools	Creating an account when you run the air-gapped install scripts
Red Hat® OpenShift® Container Platform CLI (oc)	Red Hat OpenShift Container Platform administration
IBM Catalog Management Plug-in for IBM Cloud Pak	Running CASE commands

Complete the following steps on your host:

1. Install OpenSSL version 1.11.1 or higher.
2. Install Docker or Podman.
   - To install Podman, see Podman Installation Instructions.
   - To install Docker (for example, on Red Hat® Enterprise Linux®), run the following commands:

     ```
     yum check-update
     yum install docker
     ```

     Note: Docker is not shipped or supported by Red Hat for Red Hat Enterprise Linux (RHEL) 8. The podman container engine replaced docker as the preferred, maintained, and supported container runtime of choice for Red Hat Enterprise Linux 8 systems. For more information, see the Red Hat documentation Running containers without Docker.
3. Install `httpd-tools`.

   ```
   yum install httpd-tools
   ```
4. Install the `oc` Red Hat® OpenShift® Container Platform CLI tool.

   `oc` is required for Red Hat OpenShift management. For more information, see Getting started with the OpenShift CLI in the Red Hat OpenShift documentation.

1.4 Install the IBM Catalog Management Plug-in for IBM Cloud Pak®

The IBM Catalog Management Plug-in for IBM Cloud Pak (**ibm-pak-plugin**) is used for the deployment of IBM Cloud Paks® in a disconnected environment. It simplifies the process for discovering required IBM product images and uses standard tools for registry and cluster access. The ibm-pak-plugin also extends the Red Hat OpenShift CLI (**oc**) capability to streamline the process of delivering installation images to the IBM Cloud Pak in an air-gapped environment.

1. Download and install the most recent version of the **ibm-pak-plugin** for your host operating system from github.com/IBM.

2. Run the following command to extract the files.

   ```
   tar -xf oc-ibm_pak-linux-amd64.tar.gz
   ```

3. Run the following command to move the file to the **/usr/local/bin** directory.

   ```
   mv oc-ibm_pak-linux-amd64 /usr/local/bin/oc-ibm_pak
   ```

   Note: If you are installing as a non-root user, you must use sudo.

4. Confirm that the **ibm-pak-plugin** is installed by running the following command.

   ```
   oc ibm-pak --help
   ```

   Expected result: The **ibm-pak-plugin** usage is displayed.

1.5 Download documentation and scripts for offline access

Download the following documents and scripts that you will need to be able to access offline during your Infrastructure Automation installation or uninstallation, and copy them to your air-gapped environment.

Cloud Pak for AIOps 4.2.0 documentation

Download the Cloud Pak for AIOps 4.2.0 documentation (this documentation) so that you can access it offline. You can download a PDF of the Cloud Pak for AIOps Infrastructure Automation documentation from PDF documentation.

Red Hat OpenShift Container Platform documentation

The Red Hat OpenShift Container Platform documentation can be downloaded for offline access from Red Hat.

Download the following PDFs:

- Red Hat OpenShift Container Platform 4.10 Operators PDF.
- Red Hat Operators Container Platform 4.10 Installing PDF.

IBM Cloud Paks® documentation

Download the IBM Cloud Paks® documentation so that you can access it offline. You can download a PDF of the documentation from PDF documentation.

2. Set environment variables and download CASE files

Before mirroring your images, set environment variables on your mirroring device, and connect to the internet so that you can download the corresponding CASE files. To finish preparing your host, complete the following steps.

Note: Save a copy of your environment variable values to a text editor. You can use that file as a reference to cut and paste from when completing your air-gapped installation tasks.

1. Create the following environment variables with the installer image name and the image inventory on your host.

   ```
   export IA_CASE_NAME=ibm-ia-installer
   export IA_CASE_VERSION=1.6.0
   export IA_CASE_ARCHIVE=$IA_CASE_NAME-$IA_CASE_VERSION.tgz
   export IA_CASE_INVENTORY_SETUP=ibmInfrastructureAutomationOperatorSetup
   export IA_CASE_REPO_PATH=https://github.com/IBM/cloud-pak/raw/master/repo/case
   export TARGET_REGISTRY_HOST=<IP_or_FQDN_of_local_docker_registry>
   export TARGET_REGISTRY_PORT=<port_number_of_local_docker_registry>
   export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
   ```

```
export TARGET_REGISTRY_USER=<username>
export TARGET_REGISTRY_PASSWORD=<password>
```

The target registry is the registry where the Infrastructure Automation images will be mirrored to, and accessed from by the Red Hat OpenShift cluster, as setup in 1.1 Set up a target registry.

If your bastion host must connect to the internet through a proxy, then also set the following environment variables.

```
export https_proxy=http://proxy-server-hostname:port
export http_proxy=http://proxy-server-hostname:port
```

2. Connect your host to the internet and disconnect it from the local air-gapped network.
3. Download the Infrastructure Automation installer and image inventory to your host.

```
oc ibm-pak get $IA_CASE_NAME --version $IA_CASE_VERSION
```

The CASE is downloaded to **~/.ibm-pak/data/cases/$IA_CASE_NAME/$IA_CASE_VERSION**. The logs files are available at **~/.ibm-pak/logs/oc-ibm_pak.log**.

Note: If you do not specify the CASE version, then the latest CASE is downloaded. The root directory that is used by **ibm-pak-plugin** is **~/.ibm-pak**. If required, the root directory can be configured by setting the **IBMPAK_HOME** environment variable.

4. Verify that the images are downloaded by running **ls** against your offline directory.

```
ls ~/.ibm-pak/data/cases/${IA_CASE_NAME}/${IA_CASE_VERSION}
```

Your host is now configured and you are ready to mirror your images.

3. Mirror images to your final location

The process of mirroring images takes the image from the internet to your host, then effectively copies that image on to your air-gapped environment. After you mirror your images, you can configure your cluster and complete an air-gapped installation.

Notes:

- If you want to install subsequent updates to your air-gapped environment, you must do a CASE save to get the image list when performing those updates.
- For details on ensuring you do not introduce changes to multiple environments, see Setting up a repeatable air-gap process.
- For your mirror registry certs, include a SAN entry. In addition, configure your cluster with the certs or mark the registry as insecure. For more information about how to complete the configuration, see the Red Hat OpenShift documentation Setting up additional trusted certificate authorities for builds.

Complete the following steps to mirror your images from your host to your air-gapped environment:

- 3.1 Generate mirror manifests
- 3.2 Authenticate with the registries
- 3.3 Mirror images to the host
- 3.4 Connect your host to your air-gapped environment and set up your container
- 3.5 Mirror images to final location and configure the cluster

3.1. Generate mirror manifests

Run the following command to generate mirror manifests to be used when mirroring the images to the target registry.

```
oc ibm-pak generate mirror-manifests $IA_CASE_NAME $TARGET_REGISTRY --version $IA_CASE_VERSION
```

A new directory named **~/.ibm-pak/mirror** is created when you issue the **oc ibm-pak generate mirror-manifests** command. The files **images-mapping.txt** and **image-content-source-policy.yaml** are generated at **~/.ibm-pak/data/mirror/$IA_CASE_NAME/$IA_CASE_VERSION**.

3.2. Authenticate with the registries

Log in to the registries to generate an authentication file containing the registry credentials, and then create an environment variable that has the location of the authentication file. This file is used later to enable the **oc image mirror** command to pull the images from the IBM Entitled Registry, and push them to the target registry.

1. Get the authentication credentials for the IBM Entitled Registry.

    1. To obtain the entitlement key that is assigned to your IBMid, log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.

    2. In the *Entitlement keys* section, select **Copy key** to copy the entitlement key.

2. Run the following command to create an environment variable containing your entitlement key.

    ```
    export ENTITLED_REGISTRY_PASSWORD=<key>
    ```

    Where **<key>** is the entitlement key that you copied in the previous step.

3. Store the authentication credentials for the IBM Entitled Registry and the target registry.

    If you are using Podman, run the following commands:

    ```
    podman login cp.icr.io -u cp -p $ENTITLED_REGISTRY_PASSWORD
    podman login $TARGET_REGISTRY -u $TARGET_REGISTRY_USER -p $TARGET_REGISTRY_PASSWORD
    export REGISTRY_AUTH_FILE=${XDG_RUNTIME_DIR}/containers/auth.json
    unset ENTITLED_REGISTRY_PASSWORD
    ```

    Note: The authentication file is usually at **${XDG_RUNTIME_DIR}/containers/auth.json**. For more information, see the Options section in the Podman documentation.

    If you are using Docker, run the following commands:

    ```
    docker login cp.icr.io -u cp -p $ENTITLED_REGISTRY_PASSWORD
    docker login $TARGET_REGISTRY -u $TARGET_REGISTRY_USER -p $TARGET_REGISTRY_PASSWORD
    export REGISTRY_AUTH_FILE=$HOME/.docker/config.json
    unset ENTITLED_REGISTRY_PASSWORD
    ```

    Note: The authentication file is usually at **$HOME/.docker/config.json**. For more information, see the Docker documentation.

3.3. Mirror the images

Complete these steps on your host that is connected to both the local Docker or Podman registry and the Red Hat OpenShift cluster.

Mirror images to the target registry.

```
nohup oc image mirror \
 -f ~/.ibm-pak/data/mirror/$IA_CASE_NAME/$IA_CASE_VERSION/images-mapping.txt \
 -a $REGISTRY_AUTH_FILE \
 --filter-by-os '.*' \
 --insecure \
 --skip-multiple-scopes \
 --max-per-registry=1 \
 --continue-on-error=true > my-mirror-progress.txt 2>&1 &
```

The UNIX command **nohup** is used to ensure that the mirroring process continues even if there is a loss of network connection, and redirection of output to a file provides improved monitoring and error visibilty.

Run the following command if you want to see the progress of the mirroring:

```
tail -f my-mirror-progress.txt
```

Note: If an error occurs during mirroring, the mirror command can be re-run.

3.4 Configure the cluster

1. Log in to your OpenShift cluster.

You can identify your specific **`oc login`** command by clicking the user menu in the upper left corner of the Red Hat OpenShift console, and then clicking **Copy Login Command**.

Example:

```
oc login <server> -u <cluster username> -p <cluster pass>
```

2. Update the global image pull secret for your Red Hat OpenShift cluster.

   Follow the steps in the Red Hat OpenShift documentation topic Updating the global cluster pull secret.

   These steps enable your cluster to have authentication credentials in place to pull images from your TARGET_REGISTRY as specified in the **`image-content-source-policy.yaml`**, which you will apply to your cluster in the next step.

3. Create the **`ImageContentSourcePolicy`**.

   Run the following command to create the ImageContentSourcePolicy.

   ```
   oc apply -f  ~/.ibm-pak/data/mirror/$IA_CASE_NAME/$IA_CASE_VERSION/image-content-source-policy.yaml
   ```

4. Verify that the **`ImageContentSourcePolicy`** resource is created.

   ```
   oc get imageContentSourcePolicy
   ```

5. Verify your cluster node status.

   ```
   oc get MachineConfigPool -w
   ```

   Important: After the **`ImageContentsourcePolicy`** and global image pull secret are applied, the configuration of your nodes will be updated sequentially. Wait until all of the **`MachineConfigPools`** are updated before you proceed to the next step.

6. (Optional) If you use an insecure registry, you must add the target registry to the cluster's **`insecureRegistries`** list.

   ```
   oc patch image.config.openshift.io/cluster --type=merge \
   -p '{"spec":{"registrySources":{"insecureRegistries":["'${TARGET_REGISTRY}'"]}}}'
   ```

4. Configure storage

The storage configuration must satisfy your sizing requirements. Two storage classes are needed for installing Infrastructure Automation. For more information, see Storage.

5. Install Infrastructure Automation by way of Red Hat OpenShift Container Platform

Now that your images are mirrored to your air-gapped environment, you can deploy Infrastructure Automation to that environment. When you mirrored your environment, you created a parallel offline version of everything that you needed to install an operator into Red Hat® OpenShift® Container Platform. To install Infrastructure Automation, complete the following steps:

5.1 Create a custom project (namespace)

Create a project (namespace) called **`cp4aiops`** for your Infrastructure Automation deployment, by running the following command:

```
oc create namespace cp4aiops
```

5.2 Create the catalog source

1. Set the project (namespace) to install the catalog:

   ```
   export CATALOG_NAMESPACE=openshift-marketplace
   ```

2. Create the catalog source.

```
oc ibm-pak launch \
  $IA_CASE_NAME \
  --version $IA_CASE_VERSION \
  --action install-catalog \
  --inventory $IA_CASE_INVENTORY_SETUP \
  --namespace $CATALOG_NAMESPACE \
  --args "--registry $TARGET_REGISTRY --recursive \
  --inputDir ~/.ibm-pak/data/cases/$IA_CASE_NAME/$IA_CASE_VERSION"
```

5.3 Create an air-gapped instance of Infrastructure Automation

Follow the instructions in Online production install of IBM Cloud Pak for AIOps Infrastructure Automation from step 7, *Install the Infrastructure Automation operator* onwards if you are installing Infrastructure Automation for use with IBM Cloud Pak for AIOps, or from step 6 *Create an EgressFirewall* onwards if you are installing stand-alone Infrastructure Automation.

Preparing to install IBM Cloud Pak for AIOps Infrastructure Automation in an air-gapped environment (offline) using a portable compute device

If your cluster is not connected to the internet, you can deploy a production install of IBM Cloud Pak® for AIOps Infrastructure Automation on your cluster by using a portable compute device.

In this scenario, your air-gapped (offline) environment has a target registry, and a Red Hat® OpenShift® Container Platform cluster on which Infrastructure Automation is to be installed. Infrastructure Automation images are mirrored from the internet to a file system on a portable compute device. The portable device is then disconnected from the internet and connected in the offline environment, where the images are loaded to the target registry. Infrastructure Automation can then be installed in the offline environment by using the target registry.

Notes: The following procedure is based on a Red Hat® OpenShift® Container Platform 4.12 environment and includes links for that version. If your environment uses a different supported version of Red Hat® OpenShift® Container Platform, ensure that you follow the Red Hat OpenShift documentation for that version.

Before you begin

Review the Planning section. Your environment must meet the system requirements.

Important: The **storageClass** and **storageClassLargeBlock** that are used for creating the IBM Cloud Pak for AIOps custom resource must have the same value as the **storageClass** and **storageClassLargeBlock** that are used for creating the Infrastructure Automation custom resource.

Procedure

From a high level, an air-gapped installation of Infrastructure Automation consists of five steps:

1. Set up your mirroring environment
2. Set environment variables and download CASE files
3. Mirror images to portable registry
4. Configure storage
5. Install Infrastructure Automation by way of Red Hat OpenShift Container Platform

1. Set up your mirroring environment

Before you install any IBM Cloud Pak® on an air-gapped environment, you must set up a host that can be connected to the internet to complete configuring your mirroring environment. To set up your mirroring environment, complete the following steps:

Prerequisites

You must satisfy the following prerequisites:

- Red Hat® OpenShift® Container Platform requires you to have cluster admin access to run the `configure-cluster-airgap` and `install-operator` commands.
- Allow access to the following sites and ports:

Table 1. Sites and ports that must be accessible

Site	Description
• `icr.io` • `cp.icr.io` • `dd0.icr.io` • `dd2.icr.io` • `dd4.icr.io` • `dd6.icr.io`	Allow access to these hosts on port 443 to enable access to the `IBM Cloud Container Registry`, CASE OCI artifact, and IBM Cloud Pak® foundational services catalog source.
• `dd1-icr.ibm-zh.com` • `dd3-icr.ibm-zh.com` • `dd5-icr.ibm-zh.com` • `dd7-icr.ibm-zh.com`	If you are located in China, also allow access to these hosts on port 443.
`github.com`	Github houses CASE files, IBM Cloud Pak tools and scripts.
`redhat.com`	Red Hat OpenShift registries that are required for Configuring your firewall for OpenShift Container Platform and for Red Hat OpenShift upgrades.

1.1 Install and configure Red Hat OpenShift

Infrastructure Automation requires Red Hat OpenShift to be installed and running on your target cluster. You must have administrative access to your Red Hat OpenShift cluster.

For more information about the supported versions of Red Hat OpenShift, see Supported Red Hat OpenShift Container Platform versions.

1. Install Red Hat OpenShift by using the instructions in the Red Hat OpenShift documentation. Information about installing a cluster in a restricted network is given in Mirroring images for a disconnected installation.
2. Install the Red Hat OpenShift command line interface (`oc`) on your cluster's boot node and run `oc login`. For more information, see the instructions in Getting started with the Red Hat OpenShift CLI.
3. Optionally configure a custom certificate for Infrastructure Automation to use. You can use either of the following methods:
    - Configure a custom certificate for the Red Hat OpenShift cluster. Follow the instructions in the Red Hat OpenShift documentation Replacing the default ingress certificate. Then, deploy the signing CA certificate into the cluster by following the instructions in the Red Hat OpenShift documentation Replacing the CA Bundle certificate.
    - If you would like to use a custom certificate for Infrastructure Automation only, then after installation is complete follow the instructions in Using a custom certificate.

1.2 Set up a target registry

You must have a local Docker type production-grade registry available in the air-gapped environment to store the Infrastructure Automation images in. The registry must meet the following requirements:

- supports Docker Manifest V2, Schema 2.
- supports multi-architecture images.
- is accessible from the Red Hat OpenShift cluster nodes.
- allows path separators in the image name.
- you have the username and password for a user who can read from and write to the registry.
- must have 58 GB of storage to hold all of the software that is to be transferred to the target registry.

If you do not already have a suitable production-grade registry available, then you must install and configure one. For more information, see About the mirror registry in the Red Hat OpenShift documentation.

Important: Do not use the Red Hat OpenShift image registry as your target registry. The Red Hat OpenShift registry does not support multi-architecture images or path separators in the image name.

1.3 Prepare a host

Regardless of which type of host you're using, you must be able to connect it to the internet and to the air-gapped network with access to the Red Hat® OpenShift® Container Platform cluster and the local, intranet Docker type registry. Your host must be on a Linux® x86_64 or Mac platform with any operating system that the IBM Cloud Pak® CLI and the Red Hat® OpenShift® Container Platform CLI support. If you are on a Windows platform, you must run the actions in a Linux® x86_64 VM or from a Windows Subsystem for Linux (WSL) terminal.

Your host must have 58 GB of storage to hold all of the software that is to be transferred to the local, intranet Docker type registry.

The following table explains the software requirements for installing Infrastructure Automation in an air-gapped environment:

Table 2. Software requirements and purpose

Software	Purpose
OpenSSL	Validating certificates when you run the air-gapped install scripts
Docker	Container management
Podman	Container management
Apache httpd-tools	Creating an account when you run the air-gapped install scripts
Red Hat® OpenShift® Container Platform CLI (oc)	Red Hat OpenShift Container Platform administration
IBM Catalog Management Plug-in for IBM Cloud Pak	Running CASE commands

Complete the following steps on your host:

1. Install OpenSSL version 1.11.1 or higher.
2. Install Docker or Podman.
   - To install Docker (for example, on Red Hat® Enterprise Linux®), run the following commands:

     ```
     yum check-update
     yum install docker
     ```

     Note: Docker is not shipped or supported by Red Hat for Red Hat Enterprise Linux (RHEL) 8. The podman container engine replaced docker as the preferred, maintained, and supported container runtime of choice for Red Hat Enterprise Linux 8 systems. For more information, see the Red Hat documentation Running containers without Docker.
   - To install Podman, see Podman Installation Instructions.
3. Install `httpd-tools`.

   ```
   yum install httpd-tools
   ```
4. Install the `oc` Red Hat® OpenShift® Container Platform CLI tool. For more information, see Install the OpenShift CLI.
5. Create a directory that serves as the offline store.

   Following is an example directory, which is used in the subsequent steps.

```
mkdir $HOME/offline
```

Note: This offline store must be persistent to avoid transferring data more than once. The persistence also helps to run the mirroring process multiple times or on a schedule.

The portable device must be configured such that it can independently connect to the internet and to the air-gapped network.

1.4 Install the IBM Catalog Management Plug-in for IBM Cloud Pak®

The IBM Catalog Management Plug-in for IBM Cloud Pak (**ibm-pak-plugin**) is used for the deployment of IBM Cloud Paks® in a disconnected environment. It simplifies the process for discovering required IBM product images and uses standard tools for registry and cluster access. The ibm-pak-plugin also extends the Red Hat OpenShift CLI (**oc**) capability to streamline the process of delivering installation images to the IBM Cloud Pak in an air-gapped environment.

1. Download and install the most recent version of the **ibm-pak-plugin** for your host operating system from github.com/IBM.
2. Run the following command to extract the files.

   ```
   tar -xf oc-ibm_pak-linux-amd64.tar.gz
   ```

3. Run the following command to move the file to the **/usr/local/bin** directory.

   ```
   mv oc-ibm_pak-linux-amd64 /usr/local/bin/oc-ibm_pak
   ```

   Note: If you are installing as a non-root user, you must use sudo.

4. Confirm that the **ibm-pak-plugin** is installed by running the following command.

   ```
   oc ibm-pak --help
   ```

   Expected result: The **ibm-pak-plugin** usage is displayed.

1.5 Download documentation and scripts for offline access

Download the following documents and scripts that you will need to be able to access offline during your Infrastructure Automation installation or uninstallation, and copy them to your air-gapped environment.

Cloud Pak for AIOps 4.2.0 documentation

Download the Cloud Pak for AIOps 4.2.0 documentation (this documentation) so that you can access it offline. You can download a PDF of the Cloud Pak for AIOps Infrastructure Automation documentation from PDF documentation.

Red Hat OpenShift Container Platform documentation

The Red Hat OpenShift documentation can be downloaded for offline access from Red Hat. The **CLI tools**, **Images**, **Operators**, and **Networking** sections are referenced by this documentation.

IBM Cloud Paks® documentation

Download the IBM Cloud Paks® documentation so that you can access it offline. You can download a PDF of the documentation from PDF documentation.

2. Set environment variables and download CASE files

Before mirroring your images, you can set the environment variables on your mirroring device, and connect to the internet so that you can download the corresponding CASE files. To finish preparing your host, complete the following steps:

Note: Save a copy of your environment variable values to a text editor. You can use that file as a reference to cut and paste from when completing your air-gapped installation tasks.

1. Create the following environment variables with the installer image name and the image inventory on your host.

   ```
   export IA_CASE_NAME=ibm-ia-installer
   export IA_CASE_VERSION=1.6.0
   export IA_CASE_ARCHIVE=$IA_CASE_NAME-$IA_CASE_VERSION.tgz
   ```

```
export IA_CASE_INVENTORY_SETUP=ibmInfrastructureAutomationOperatorSetup
export OFFLINEDIR_ARCHIVE=offline.tgz
export IA_CASE_REPO_PATH=https://github.com/IBM/cloud-pak/raw/master/repo/case
export TARGET_REGISTRY_HOST=<IP_or_FQDN_of_target_registry>
export TARGET_REGISTRY_PORT=<port_number_of_target_registry>
export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
export TARGET_REGISTRY_USER=<username>
export TARGET_REGISTRY_PASSWORD=<password>
export TARGET_REGISTRY_PATH=$OFFLINEDIR/imageregistry
```

The target registry is the registry where the Infrastructure Automation images will be mirrored to, and accessed from by the Red Hat OpenShift cluster, as setup in 1.2 Set up a target registry

2. Connect your host to the internet and disconnect it from the local air-gapped network.
3. Download the Infrastructure Automation installer and image inventory to your host.

```
oc ibm-pak get $IA_CASE_NAME --version $IA_CASE_VERSION
```

The CASE is downloaded to **~/.ibm-pak/data/cases/$IA_CASE_NAME/$IA_CASE_VERSION**. The logs files are available at **~/.ibm-pak/logs/oc-ibm_pak.log**.

Note: If you do not specify the CASE version, then the latest CASE is downloaded. The root directory that is used by **ibm-pak-plugin** is **~/.ibm-pak**. If required, the root directory can be configured by setting the **IBMPAK_HOME** environment variable.

4. Verify that the images are downloaded by running **ls** against your offline directory.

```
ls ~/.ibm-pak/data/cases/${IA_CASE_NAME}/${IA_CASE_VERSION}
```

Your host is now configured and you are ready to mirror your images.

3. Mirror images to portable registry

The process of mirroring images takes the image from the internet to your host, then effectively copies that image on to your air-gapped environment. After you mirror your images, you can configure your cluster and complete an air-gapped installation.

Notes:

- If you want to install subsequent updates to your air-gapped environment, you must do a CASE save to get the image list when performing those updates.
- For details on ensuring you do not introduce changes to multiple environments, see Setting up a repeatable air-gap process.
- For your mirror registry certs, include a SAN entry. In addition, configure your cluster with the certs or mark the registry as insecure. For more information about how to complete the configuration, see the Red Hat OpenShift documentation Setting up additional trusted certificate authorities for builds.

Complete the following steps to mirror your images from your host to your air-gapped environment:

3.1. Generate mirror manifests

Run the following command to generate mirror manifests to be used when mirroring the images to the target registry.

```
oc ibm-pak generate mirror-manifests $IA_CASE_NAME file://cpfs --version $IA_CASE_VERSION --
final-registry $TARGET_REGISTRY/cpfs
```

A new directory named **~/.ibm-pak/mirror** is created when you issue the **oc ibm-pak generate mirror-manifests** command. The files **images-mapping.txt** and **image-content-source-policy.yaml** are generated at **~/.ibm-pak/data/mirror/$IA_CASE_NAME/$IA_CASE_VERSION**.

Table 3. Parameter description

Argument	Description
file://cpfs	Specifies the path extension where images will be mirrored to. Images are mirrored to **$IMAGE_PATH/cpfs** when the **oc image mirror** command is run with **images-mapping-to-filesystem.txt**. For more information, see Mirror the images to the file system.

Argument	Description
`$TARGET_REGISTRY/cpfs`	This argument generates an image-mapping file that is used by the `oc image mirror` commands to mirror images to the `TARGET_REGISTRY` at namespace `cpfs`. For example, if the CASE you are installing needs the image, `quay.io/opencloudio/ibm-events-kafka-2.6.0@sha256:10d422dddd29ff19c87066fc6510eee05f5fa4ff608b87a06e898b3b6a3a13c7`, then its final URL in your target registry will be `$TARGET_REGISTRY/cpfs/opencloudio/ibm-events-kafka-2.6.0`. Note the new namespace of `cpfs` in the URL. The namespace path can be multi level if your target registry supports it.

The command will generate the following files at `~/.ibm-pak/data/mirror/$IA_CASE_NAME/$IA_CASE_VERSION`:

- images-mapping-to-filesystem.txt
- images-mapping-from-filesystem.txt
- image-content-source-policy.yaml

3.2. Authenticate with the registries

Log in to the registries to generate an authentication file containing the registry credentials, and then create an environment variable that has the location of the authentication file. This file is used later to enable the `oc image mirror` command to pull the images from the IBM Entitled Registry, and push them to the target registry.

1. Get the authentication credentials for the IBM Entitled Registry.

    1. To obtain the entitlement key that is assigned to your IBMid, log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.

    2. In the *Entitlement keys* section, select **Copy key** to copy the entitlement key.

2. Run the following command to create an environment variable containing your entitlement key.

    ```
    export ENTITLED_REGISTRY_PASSWORD=<key>
    ```

    Where `<key>` is the entitlement key that you copied in the previous step.

3. Store the authentication credentials for the IBM Entitled Registry and the target registry.

    If you are using Podman, run the following commands:

    ```
    podman login cp.icr.io -u cp -p $ENTITLED_REGISTRY_PASSWORD
    podman login $TARGET_REGISTRY -u $TARGET_REGISTRY_USER -p $TARGET_REGISTRY_PASSWORD
    export REGISTRY_AUTH_FILE=${XDG_RUNTIME_DIR}/containers/auth.json
    unset ENTITLED_REGISTRY_PASSWORD
    ```

    Note: The authentication file is usually at `${XDG_RUNTIME_DIR}/containers/auth.json`. For more information, see the Options section in the Podman documentation.

    If you are using Docker, run the following commands:

    ```
    docker login cp.icr.io -u cp -p $ENTITLED_REGISTRY_PASSWORD
    docker login $TARGET_REGISTRY -u $TARGET_REGISTRY_USER -p $TARGET_REGISTRY_PASSWORD
    export REGISTRY_AUTH_FILE=$HOME/.docker/config.json
    unset ENTITLED_REGISTRY_PASSWORD
    ```

    Note: The authentication file is usually at `$HOME/.docker/config.json`. For more information, see the Docker documentation.

3.3. Mirror the images to the file system

Complete these steps to mirror the images from the internet to a file system on your portable device.

1. Create an environment variable to store the location of the file system where the images are to be stored.

    ```
    export IMAGE_PATH=<image-path>
    ```

    Where `<image-path>` is the directory where you want the images to be stored.

2. Run the following command to mirror the images from the IBM Entitled Registry to the file system.

```
nohup oc image mirror \
-f ~/.ibm-pak/data/mirror/$IA_CASE_NAME/$IA_CASE_VERSION/images-mapping-to-filesystem.txt
\
-a $REGISTRY_AUTH_FILE \
--filter-by-os '.*' \
--insecure \
--skip-multiple-scopes \
--dir "$IMAGE_PATH" \
--max-per-registry=1 > my-mirror-progress.txt 2>&1 &
```

The UNIX command **nohup** is used to ensure that the mirroring process continues even if there is a loss of network connection, and redirection of output to a file provides improved monitoring and error visibilty.

Run the following command if you want to see the progress of the mirroring:

```
tail -f my-mirror-progress.txt
```

Note: If an error occurs during mirroring, the mirror command can be re-run.

3.4 Setup the file system in the air-gapped environment

1. Disconnect the device which has your file system, (the portable compute device or the local compute device) from the internet and connect it to the air-gapped environment.
2. Ensure that environment variables are set on the device in the air-gapped environment that has access to the target registry.

   If you are using a portable storage device, then set the following environment variables on your local compute device within the air-gapped environment.

   If you are using a portable compute device which you have restarted since mirroring the images, then your environment variables will have been lost and you will need to set the following environment variables on your portable compute device again.

   ```
   export CASE_NAME=ibm-cp-waiops
   export CASE_VERSION=1.6.0
   export TARGET_REGISTRY_HOST=<IP_or_FQDN_of_target_registry>
   export TARGET_REGISTRY_PORT=<port_number_of_target_registry>
   export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
   export TARGET_REGISTRY_USER=<username>
   export TARGET_REGISTRY_PASSWORD=<password>
   export IMAGE_PATH=<image_path>
   ```

3.5 Authenticate with the target registry

Authenticate with the target registry in the air-gapped environment that you will be mirroring the images into.

If you are using Podman, run the following command:

```
podman login $TARGET_REGISTRY -u $TARGET_REGISTRY_USER -p $
export REGISTRY_AUTH_FILE=${XDG_RUNTIME_DIR}/containers/auth.json
```

Note: The authentication file is usually at **${XDG_RUNTIME_DIR}/containers/auth.json**. For more information, see the Options section in the Podman documentation.

If you are using Docker, run the following command:

```
docker login $TARGET_REGISTRY -u $TARGET_REGISTRY_USER -p $TARGET_REGISTRY_PASSWORD
export REGISTRY_AUTH_FILE=$HOME/.docker/config.json
```

Note: The authentication file is usually at **$HOME/.docker/config.json**. For more information, see the Docker documentation.

3.6 Mirror the images to the target registry from the file system

Complete the steps in this section on the device which has your file system (the portable compute device or the local compute device) to copy the images from the file system to the $TARGET_REGISTRY. Your device with the file system must be connected to both the target registry and the Red Hat OpenShift cluster.

Run the following command to copy the images referenced in the **images-mapping-from-filesystem.txt** from the $IMAGE_PATH file system to the final target registry.

```
nohup oc image mirror \
-f ~/.ibm-pak/data/mirror/$IA_CASE_NAME/$IA_CASE_VERSION/images-mapping-from-filesystem.txt \
--from-dir "$IMAGE_PATH" \
-a $REGISTRY_AUTH_FILE \
--filter-by-os '.*' \
--insecure  \
--skip-multiple-scopes \
--max-per-registry=1 > my-mirror-progress2.txt 2>&1 &
```

The UNIX command **nohup** is used to ensure that the mirroring process continues even if there is a loss of network connection, and redirection of output to a file provides improved monitoring and error visibilty.

Run the following command if you want to see the progress of the mirroring:

```
tail -f my-mirror-progress2.txt
```

Note: If an error occurs during mirroring, the mirror command can be re-run.

3.7 Configure the cluster

1. Log in to your OpenShift cluster.

   You can identify your specific **oc login** command by clicking the user menu in the upper left corner of the Red Hat OpenShift console, and then clicking **Copy Login Command**.

   Example:

   ```
   oc login <server> -u <cluster username> -p <cluster pass>
   ```

2. Update the global image pull secret for your Red Hat OpenShift cluster.

   Follow the steps in the Red Hat OpenShift documentation topic Updating the global cluster pull secret.

   These steps enable your cluster to have authentication credentials in place to pull images from your TARGET_REGISTRY as specified in the **image-content-source-policy.yaml**, which you will apply to your cluster in the next step.

3. Create the **ImageContentSourcePolicy**.

   Run the following command to create an ImageContentSourcePolicy (ICSP).

   ```
   oc apply -f  ~/.ibm-pak/data/mirror/$IA_CASE_NAME/$IA_CASE_VERSION/image-content-source-policy.yaml
   ```

4. Verify that the **ImageContentSourcePolicy** resource is created.

   ```
   oc get imagecontentsourcepolicy
   ```

5. Verify your cluster node status.

   ```
   oc get MachineConfigPool -w
   ```

   Important: After the **ImageContentsourcePolicy** and global image pull secret are applied, the configuration of your nodes will be updated sequentially. Wait until all of the **MachineConfigPools** are updated before proceeding to the next step.

6. (Optional) If you use an insecure registry, you must add the target registry to the cluster's **insecureRegistries** list.

   ```
   oc patch image.config.openshift.io/cluster --type=merge \
   -p '{"spec":{"registrySources":{"insecureRegistries":["'${TARGET_REGISTRY}'"]}}}'
   ```

4. Configure storage

The storage configuration must satisfy your sizing requirements. Two storage classes are needed for installing Infrastructure Automation. For more information, see Storage.

5. Install Infrastructure Automation by way of Red Hat OpenShift Container Platform

Now that your images are mirrored to your air-gapped environment, you can deploy Infrastructure Automation to that environment. When you mirrored your environment, you created a parallel offline version of everything that you needed to install an operator into Red Hat® OpenShift® Container Platform. To install Infrastructure Automation, complete the following steps:

5.1 Create a custom project (namespace)

Create a project (namespace) called **cp4aiops** for your Infrastructure Automation deployment, by running the following command:

```
oc create namespace cp4aiops
```

5.2 Create the catalog source

1. Set the project (namespace) to install the catalog:

   ```
   export CATALOG_NAMESPACE=openshift-marketplace
   ```

2. Create the catalog source.

   ```
   oc ibm-pak launch \
     $IA_CASE_NAME \
     --version $IA_CASE_VERSION \
     --action install-catalog \
     --inventory $IA_CASE_INVENTORY_SETUP \
     --namespace $CATALOG_NAMESPACE \
     --args "--registry $TARGET_REGISTRY --recursive \
     --inputDir ~/.ibm-pak/data/cases/$IA_CASE_NAME/$IA_CASE_VERSION"
   ```

5.3 Create an air-gapped instance of Infrastructure Automation

Follow the instructions in Online production install of IBM Cloud Pak for AIOps Infrastructure Automation from step 7, *Install the Infrastructure Automation operator* onwards if you are installing Infrastructure Automation for use with IBM Cloud Pak for AIOps, or from step 6 *Create an EgressFirewall* onwards if you are installing stand-alone Infrastructure Automation.

Preparing to install IBM Cloud Pak for AIOps Infrastructure Automation in an air-gapped environment (offline) using a portable storage device

If your cluster is not connected to the internet, you can deploy an installation of IBM Cloud Pak® for AIOps Infrastructure Automation on your cluster by using a portable storage device such as a USB device.

In this scenario, your air-gapped (offline) environment has a target registry, and a Red Hat® OpenShift® Container Platform cluster on which Infrastructure Automation is to be installed. Infrastructure Automation images are mirrored from the internet to a file system on a portable storage device. The portable device is then disconnected from the internet and connected in the offline environment, where the images are loaded to the target registry. Infrastructure Automation can then be installed in the offline environment by using the target registry.

Note: The following procedure is based on a Red Hat® OpenShift® Container Platform 4.12 environment and includes links for that version. If your environment uses a different supported version of Red Hat® OpenShift® Container Platform, ensure that you follow the Red Hat OpenShift documentation for that version.

Before you begin

Review the Planning section. Your environment must meet the system requirements.

Important: The `storageClass` and `storageClassLargeBlock` that are used for creating the IBM Cloud Pak for AIOps custom resource must have the same value as the `storageClass` and `storageClassLargeBlock` that are used for creating the Infrastructure Automation custom resource.

Procedure

From a high level, an air-gapped installation of Infrastructure Automation consists of five steps:

1. Set up your mirroring environment
2. Set environment variables and download CASE files
3. Mirror images to portable registry
4. Configure storage
5. Install Infrastructure Automation by way of Red Hat OpenShift Container Platform

1. Set up your mirroring environment

Before you install any IBM Cloud Pak® on an air-gapped environment, you must set up a host that can be connected to the internet to complete configuring your mirroring environment. To set up your mirroring environment, complete the following steps:

Prerequisites

You must satisfy the following prerequisites:

- Red Hat® OpenShift® Container Platform requires you to have cluster admin access to run the `configure-cluster-airgap` and `install-operator` commands.
- The connected compute device that will be used to access the internet and copy images to the portable storage device must have access to the following sites and ports:

Table 1. Sites and ports that must be accessible

Site	Description
• `icr.io` • `cp.icr.io` • `dd0.icr.io` • `dd2.icr.io` • `dd4.icr.io` • `dd6.icr.io`	Allow access to these hosts on port 443 to enable access to the `IBM Cloud Container Registry`, CASE OCI artifact, and IBM Cloud Pak® foundational services catalog source.
• `dd1-icr.ibm-zh.com` • `dd3-icr.ibm-zh.com` • `dd5-icr.ibm-zh.com` • `dd7-icr.ibm-zh.com`	If you are located in China, also allow access to these hosts on port 443.
`github.com`	Github houses CASE files, IBM Cloud Pak tools and scripts.
`redhat.com`	Red Hat OpenShift registries that are required for Configuring your firewall for OpenShift Container Platform and for Red Hat OpenShift upgrades.

1.1 Install and configure Red Hat OpenShift

Infrastructure Automation requires Red Hat OpenShift to be installed and running on your target cluster. You must have administrative access to your Red Hat OpenShift cluster.

For more information about the supported versions of Red Hat OpenShift, see Supported Red Hat OpenShift Container Platform versions.

1. Install Red Hat OpenShift by using the instructions in the Red Hat OpenShift documentation. Information about installing a cluster in a restricted network is given in Mirroring images for a disconnected installation.
2. Install the Red Hat OpenShift command line interface (`oc`) on your cluster's boot node and run `oc login`. For more information, see the instructions in Getting started with the Red Hat OpenShift CLI.
3. Optionally configure a custom certificate for Infrastructure Automation to use. You can use either of the following methods:
    - Configure a custom certificate for the Red Hat OpenShift cluster. Follow the instructions in the Red Hat OpenShift documentation Replacing the default ingress certificate. Then, deploy the signing CA certificate into the cluster by following the instructions in the Red Hat OpenShift documentation Replacing the CA Bundle certificate.
    - If you would like to use a custom certificate for Infrastructure Automation only, then after installation is complete follow the instructions in Using a custom certificate.

1.2 Set up a target registry

You must have a local Docker type production-grade registry available in the air-gapped environment to store the Infrastructure Automation images in. The registry must meet the following requirements:

- supports Docker Manifest V2, Schema 2.
- supports multi-architecture images.
- is accessible from the Red Hat OpenShift cluster nodes.
- allows path separators in the image name.
- you have the username and password for a user who can read from and write to the registry.
- must have 58 GB of storage to hold all of the software that is to be transferred to the target registry.

If you do not already have a suitable production-grade registry available, then you must install and configure one. For more information, see About the mirror registry in the Red Hat OpenShift documentation.

Important: Do not use the Red Hat OpenShift image registry as your target registry. The Red Hat OpenShift registry does not support multi-architecture images or path separators in the image name.

1.3 Prepare a host

Regardless of which type of host you're using, you must be able to connect it to the internet and to the air-gapped network with access to the Red Hat® OpenShift® Container Platform cluster and the local, intranet Docker type registry. Your host must be on a Linux® x86_64 or Mac platform with any operating system that the IBM Cloud Pak® CLI and the Red Hat® OpenShift® Container Platform CLI support. If you are on a Windows platform, you must run the actions in a Linux® x86_64 VM or from a Windows Subsystem for Linux (WSL) terminal.

Your host must have 58 GB of storage to hold all of the software that is to be transferred to the local, intranet Docker type registry.

The following table explains the software requirements for installing Infrastructure Automation in an air-gapped environment:

Table 2. Software requirements and purpose

Software	Purpose
OpenSSL	Validating certificates when you run the air-gapped install scripts
Docker	Container management
Podman	Container management
Apache httpd-tools	Creating an account when you run the air-gapped install scripts
Red Hat® OpenShift® Container Platform CLI (oc)	Red Hat OpenShift Container Platform administration
IBM Catalog Management Plug-in for IBM Cloud Pak	Running CASE commands

Complete the following steps on your host:

1. Install OpenSSL version 1.11.1 or higher.
2. Install Docker or Podman.
    - To install Docker (for example, on Red Hat® Enterprise Linux®), run the following commands:

      ```
      yum check-update
      yum install docker
      ```

      Note: Docker is not shipped or supported by Red Hat for Red Hat Enterprise Linux (RHEL) 8. The podman container engine replaced docker as the preferred, maintained, and supported container runtime of choice for Red Hat Enterprise Linux 8 systems. For more information, see the Red Hat documentation Running containers without Docker.
    - To install Podman, see Podman Installation Instructions.
3. Install **httpd-tools**.

   ```
   yum install httpd-tools
   ```
4. Install the **oc** Red Hat® OpenShift® Container Platform CLI tool. For more information, see Install the OpenShift CLI.
5. Create a directory that serves as the offline store.

   Following is an example directory, which is used in the subsequent steps.

   ```
   mkdir $HOME/offline
   ```

   Note: This offline store must be persistent to avoid transferring data more than once. The persistence also helps to run the mirroring process multiple times or on a schedule.

The portable storage device must have sufficient storage, and must be attached to the external host.

1.4 Install the IBM Catalog Management Plug-in for IBM Cloud Pak®

The IBM Catalog Management Plug-in for IBM Cloud Pak (**ibm-pak-plugin**) is used for the deployment of IBM Cloud Paks® in a disconnected environment. It simplifies the process for discovering required IBM product images and uses standard tools for registry and cluster access. The ibm-pak-plugin also extends the Red Hat OpenShift CLI (**oc**) capability to streamline the process of delivering installation images to the IBM Cloud Pak in an air-gapped environment.

1. Download and install the most recent version of the **ibm-pak-plugin** for your host operating system from github.com/IBM.
2. Run the following command to extract the files.

   ```
   tar -xf oc-ibm_pak-linux-amd64.tar.gz
   ```
3. Run the following command to move the file to the **/usr/local/bin** directory.

   ```
   mv oc-ibm_pak-linux-amd64 /usr/local/bin/oc-ibm_pak
   ```

   Note: If you are installing as a non-root user, you must use sudo.
4. Confirm that the **ibm-pak-plugin** is installed by running the following command.

   ```
   oc ibm-pak --help
   ```

   Expected result: The **ibm-pak-plugin** usage is displayed.

1.5 Download documentation and scripts for offline access

Download the following documents and scripts that you will need to be able to access offline during your Infrastructure Automation installation or uninstallation, and copy them to your air-gapped environment.

Cloud Pak for AIOps 4.2.0 documentation

Download the Cloud Pak for AIOps 4.2.0 documentation (this documentation) so that you can access it offline. You can download a PDF of the Cloud Pak for AIOps Infrastructure Automation documentation from PDF documentation.

Red Hat OpenShift Container Platform documentation

The Red Hat OpenShift documentation can be downloaded for offline access from Red Hat. The **CLI tools**, **Images**, **Operators**, and **Networking** sections are referenced by this documentation.

IBM Cloud Paks® documentation

Download the IBM Cloud Paks® documentation so that you can access it offline. You can download a PDF of the documentation from PDF documentation.

2. Set environment variables and download CASE files

Before mirroring your images, set the environment variables on your mirroring device, and connect to the internet so that you can download the corresponding CASE files.

Note: Save a copy of your environment variable values to a text editor. You can use that file as a reference to cut and paste from when completing your air-gapped installation tasks.

To finish preparing your host, complete the following steps:

1. Create the following environment variables with the installer image name and the image inventory on your host.

   ```
   export IA_CASE_NAME=ibm-ia-installer
   export IA_CASE_VERSION=1.6.0
   export IA_CASE_ARCHIVE=$IA_CASE_NAME-$IA_CASE_VERSION.tgz
   export IA_CASE_INVENTORY_SETUP=ibmInfrastructureAutomationOperatorSetup
   export OFFLINEDIR_ARCHIVE=offline.tgz
   export IA_CASE_REPO_PATH=https://github.com/IBM/cloud-pak/raw/master/repo/case
   export TARGET_REGISTRY_HOST=<IP_or_FQDN_of_target_registry>
   export TARGET_REGISTRY_PORT=<port_number_of_target_registry>
   export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
   export TARGET_REGISTRY_USER=<username>
   export TARGET_REGISTRY_PASSWORD=<password>
   export TARGET_REGISTRY_PATH=$OFFLINEDIR/imageregistry
   ```

   The target registry is the registry where the Infrastructure Automation images will be mirrored to, and accessed from by the Red Hat OpenShift cluster, as setup in 1.2 Set up a target registry

2. Connect your host to the internet and disconnect it from the local air-gapped network.

3. Download the Infrastructure Automation installer and image inventory to your host.

   ```
   oc ibm-pak get $IA_CASE_NAME --version $IA_CASE_VERSION
   ```

   The CASE is downloaded to **~/.ibm-pak/data/cases/$IA_CASE_NAME/$IA_CASE_VERSION**. The logs files are available at **~/.ibm-pak/logs/oc-ibm_pak.log**.

   Note: If you do not specify the CASE version, then the latest CASE is downloaded. The root directory that is used by **ibm-pak-plugin** is **~/.ibm-pak**. If required, the root directory can be configured by setting the **IBMPAK_HOME** environment variable.

4. Verify that the images are downloaded by running **ls** against your offline directory.

   ```
   ls ~/.ibm-pak/data/cases/${IA_CASE_NAME}/${IA_CASE_VERSION}
   ```

Your host is now configured and you are ready to mirror your images.

3. Mirror images to portable registry

The process of mirroring images takes the image from the internet to your host, then effectively copies that image on to your air-gapped environment. After you mirror your images, you can configure your cluster and complete an air-gapped installation.

Notes:

- If you want to install subsequent updates to your air-gapped environment, you must do a CASE save to get the image list when performing those updates.

- For details on ensuring you do not introduce changes to multiple environments, see Setting up a repeatable air-gap process.
- For your mirror registry certs, include a SAN entry. In addition, configure your cluster with the certs or mark the registry as insecure. For more information about how to complete the configuration, see the Red Hat OpenShift documentation Setting up additional trusted certificate authorities for builds.

Complete the following steps to mirror your images from your host to your air-gapped environment:

3.1. Generate mirror manifests

Run the following command to generate mirror manifests to be used when mirroring the images to the target registry.

```
oc ibm-pak generate mirror-manifests $IA_CASE_NAME file://cpfs --version $IA_CASE_VERSION --final-registry $TARGET_REGISTRY/cpfs
```

A new directory named **~/.ibm-pak/mirror** is created when you issue the **oc ibm-pak generate mirror-manifests** command. The files **images-mapping.txt** and **image-content-source-policy.yaml** are generated at **~/.ibm-pak/data/mirror/$IA_CASE_NAME/$IA_CASE_VERSION**.

Table 3. Parameter description

Argument	Description
`file://cpfs`	Specifies the path extension where images will be mirrored to. Images are mirrored to **$IMAGE_PATH/cpfs** when the **oc image mirror** command is run with **images-mapping-to-filesystem.txt**. For more information, see Mirror the images to the file system.
`$TARGET_REGISTRY/cpfs`	This argument generates an image-mapping file that is used by the **oc image mirror** commands to mirror images to the **TARGET_REGISTRY** at namespace **cpfs**. For example, if the CASE you are installing needs the image, **quay.io/opencloudio/ibm-events-kafka-2.6.0@sha256:10d422dddd29ff19c87066fc6510eee05f5fa4ff608b87a06e898b3b6a3a13c7**, then its final URL in your target registry will be **$TARGET_REGISTRY/cpfs/opencloudio/ibm-events-kafka-2.6.0**. Note the new namespace of **cpfs** in the URL. The namespace path can be multi level if your target registry supports it.

The command will generate the following files at **~/.ibm-pak/data/mirror/$IA_CASE_NAME/$IA_CASE_VERSION**:

- images-mapping-to-filesystem.txt
- images-mapping-from-filesystem.txt
- image-content-source-policy.yaml

3.2. Authenticate with the registries

Log in to the registries to generate an authentication file containing the registry credentials, and then create an environment variable that has the location of the authentication file. This file is used later to enable the **oc image mirror** command to pull the images from the IBM Entitled Registry, and push them to the target registry.

1. Get the authentication credentials for the IBM Entitled Registry.
    1. To obtain the entitlement key that is assigned to your IBMid, log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.
    2. In the *Entitlement keys* section, select **Copy key** to copy the entitlement key.
2. Run the following command to create an environment variable containing your entitlement key.

    ```
    export ENTITLED_REGISTRY_PASSWORD=<key>
    ```

    Where **<key>** is the entitlement key that you copied in the previous step.
3. Store the authentication credentials for the IBM Entitled Registry and the target registry.

    If you are using Podman, run the following commands:

    ```
    podman login cp.icr.io -u cp -p $ENTITLED_REGISTRY_PASSWORD
    podman login $TARGET_REGISTRY -u $TARGET_REGISTRY_USER -p $TARGET_REGISTRY_PASSWORD
    export REGISTRY_AUTH_FILE=${XDG_RUNTIME_DIR}/containers/auth.json
    unset ENTITLED_REGISTRY_PASSWORD
    ```

Note: The authentication file is usually at `${XDG_RUNTIME_DIR}/containers/auth.json`. For more information, see the Options section in the Podman documentation.

If you are using Docker, run the following commands:

```
docker login cp.icr.io -u cp -p $ENTITLED_REGISTRY_PASSWORD
docker login $TARGET_REGISTRY -u $TARGET_REGISTRY_USER -p $TARGET_REGISTRY_PASSWORD
export REGISTRY_AUTH_FILE=$HOME/.docker/config.json
unset ENTITLED_REGISTRY_PASSWORD
```

Note: The authentication file is usually at `$HOME/.docker/config.json`. For more information, see the Docker documentation.

3.3. Mirror the images to the file system

Complete these steps to mirror the images from the internet to a file system on your portable device.

1. Create an environment variable to store the location of the file system where the images are to be stored.

   ```
   export IMAGE_PATH=<image-path>
   ```

   Where `<image-path>` is the directory where you want the images to be stored.

2. Run the following command to mirror the images from the IBM Entitled Registry to the file system.

   ```
   nohup oc image mirror \
   -f ~/.ibm-pak/data/mirror/$IA_CASE_NAME/$IA_CASE_VERSION/images-mapping-to-filesystem.txt
   \
   -a $REGISTRY_AUTH_FILE \
   --filter-by-os '.*' \
   --insecure \
   --skip-multiple-scopes \
   --dir "$IMAGE_PATH" \
   --max-per-registry=1 > my-mirror-progress.txt 2>&1 &
   ```

   The UNIX command `nohup` is used to ensure that the mirroring process continues even if there is a loss of network connection, and redirection of output to a file provides improved monitoring and error visibilty.

   Run the following command if you want to see the progress of the mirroring:

   ```
   tail -f my-mirror-progress.txt
   ```

   Note: If an error occurs during mirroring, the mirror command can be re-run.

3.4 Setup the file system in the air-gapped environment

1. Copy the files from the portable storage device to a local compute device in the air-gapped environment that has access to the target registry. If you are using a portable compute device, then these items are already present, and you can proceed to the next step.

   Copy the following items to your local compute device:

   - the file system which is located at $IMAGE_PATH, which you specified earlier
   - `~/.ibm-pak` directory

2. Disconnect the device which has your file system and connect it to the air-gapped environment.

3. Ensure that environment variables are set on the device in the air-gapped environment that has access to the target registry.

   If you are using a portable storage device, then set the following environment variables on your local compute device within the air-gapped environment.

   If you are using a portable compute device which you have restarted since mirroring the images, then your environment variables will have been lost and you will need to set the following environment variables on your portable compute device again.

   ```
   export CASE_NAME=ibm-cp-waiops
   export CASE_VERSION=1.6.0
   export TARGET_REGISTRY_HOST=<IP_or_FQDN_of_target_registry>
   ```

```
export TARGET_REGISTRY_PORT=<port_number_of_target_registry>
export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
export TARGET_REGISTRY_USER=<username>
export TARGET_REGISTRY_PASSWORD=<password>
export IMAGE_PATH=<image_path>
```

3.5 Authenticate with the target registry

Authenticate with the target registry in the air-gapped environment that you will be mirroring the images into.

If you are using Podman, run the following command:

```
podman login $TARGET_REGISTRY -u $TARGET_REGISTRY_USER -p $
export REGISTRY_AUTH_FILE=${XDG_RUNTIME_DIR}/containers/auth.json
```

Note: The authentication file is usually at `${XDG_RUNTIME_DIR}/containers/auth.json`. For more information, see the Options section in the Podman documentation.

If you are using Docker, run the following command:

```
docker login $TARGET_REGISTRY -u $TARGET_REGISTRY_USER -p $TARGET_REGISTRY_PASSWORD
export REGISTRY_AUTH_FILE=$HOME/.docker/config.json
```

Note: The authentication file is usually at `$HOME/.docker/config.json`. For more information, see the Docker documentation.

3.6 Mirror the images to the target registry from the file system

Complete the steps in this section on the device which has your file system (the portable compute device or the local compute device) to copy the images from the file system to the $TARGET_REGISTRY. Your device with the file system must be connected to both the target registry and the Red Hat OpenShift cluster.

Run the following command to copy the images referenced in the **`images-mapping-from-filesystem.txt`** from the $IMAGE_PATH file system to the final target registry.

```
nohup oc image mirror \
-f ~/.ibm-pak/data/mirror/$IA_CASE_NAME/$IA_CASE_VERSION/images-mapping-from-filesystem.txt \
--from-dir "$IMAGE_PATH" \
-a $REGISTRY_AUTH_FILE \
--filter-by-os '.*' \
--insecure  \
--skip-multiple-scopes \
--max-per-registry=1 > my-mirror-progress2.txt 2>&1 &
```

The UNIX command **`nohup`** is used to ensure that the mirroring process continues even if there is a loss of network connection, and redirection of output to a file provides improved monitoring and error visibilty.

Run the following command if you want to see the progress of the mirroring:

```
tail -f my-mirror-progress2.txt
```

Note: If an error occurs during mirroring, the mirror command can be re-run.

3.7 Configure the cluster

1. Log in to your OpenShift cluster.

   You can identify your specific **`oc login`** command by clicking the user menu in the upper left corner of the Red Hat OpenShift console, and then clicking **Copy Login Command**.

   Example:

   ```
   oc login <server> -u <cluster username> -p <cluster pass>
   ```

2. Update the global image pull secret for your Red Hat OpenShift cluster.

   Follow the steps in the Red Hat OpenShift documentation topic Updating the global cluster pull secret.

These steps enable your cluster to have authentication credentials in place to pull images from your TARGET_REGISTRY as specified in the **image-content-source-policy.yaml**, which you will apply to your cluster in the next step.

3. Create the **ImageContentSourcePolicy**.

   Run the following command to create an ImageContentSourcePolicy (ICSP).

   ```
   oc apply -f  ~/.ibm-pak/data/mirror/$IA_CASE_NAME/$IA_CASE_VERSION/image-content-source-policy.yaml
   ```

4. Verify that the **ImageContentSourcePolicy** resource is created.

   ```
   oc get imagecontentsourcepolicy
   ```

5. Verify your cluster node status.

   ```
   oc get MachineConfigPool -w
   ```

   Important: After the **ImageContentsourcePolicy** and global image pull secret are applied, the configuration of your nodes will be updated sequentially. Wait until all of the **MachineConfigPools** are updated before proceeding to the next step.

6. (Optional) If you use an insecure registry, you must add the target registry to the cluster's **insecureRegistries** list.

   ```
   oc patch image.config.openshift.io/cluster --type=merge \
   -p '{"spec":{"registrySources":{"insecureRegistries":["'${TARGET_REGISTRY}'"]}}}'
   ```

4. Configure storage

The storage configuration must satisfy your sizing requirements. Two storage classes are needed for installing Infrastructure Automation. For more information, see Storage.

5. Install Infrastructure Automation by way of Red Hat OpenShift Container Platform

Now that your images are mirrored to your air-gapped environment, you can deploy Infrastructure Automation to that environment. When you mirrored your environment, you created a parallel offline version of everything that you needed to install an operator into Red Hat® OpenShift® Container Platform. To install Infrastructure Automation, complete the following steps:

5.1 Create a custom project (namespace)

Create a project (namespace) called **cp4aiops** for your Infrastructure Automation deployment, by running the following command:

```
oc create namespace cp4aiops
```

5.2 Create the catalog source

1. Set the project (namespace) to install the catalog:

   ```
   export CATALOG_NAMESPACE=openshift-marketplace
   ```

2. Create the catalog source.

   ```
   oc ibm-pak launch \
     $IA_CASE_NAME \
     --version $IA_CASE_VERSION \
     --action install-catalog \
     --inventory $IA_CASE_INVENTORY_SETUP \
     --namespace $CATALOG_NAMESPACE \
     --args "--registry $TARGET_REGISTRY --recursive \
     --inputDir ~/.ibm-pak/data/cases/$IA_CASE_NAME/$IA_CASE_VERSION"
   ```

5.3 Create an air-gapped instance of Infrastructure Automation

Follow the instructions in Online production install of IBM Cloud Pak for AIOps Infrastructure Automation from step 7, *Install the Infrastructure Automation operator* onwards if you are installing Infrastructure Automation for use with IBM Cloud Pak for AIOps, or from step 6 *Create an EgressFirewall* onwards if you are installing stand-alone Infrastructure Automation.

Post installation (Infrastructure Automation for use with IBM Cloud Pak for AIOps)

After installing there are a number of related tasks that you should consider completing:

- Deploying Infrastructure management operand
- Deploying Infrastructure management as a containerized deployment (podified)
- Deploying Infrastructure management as a virtual machine appliance
- OIDC Configuration Steps for Infrastructure Management Appliance only

Deploying Infrastructure management operand

Infrastructure management delivers the insight, control, and automation enterprises need to address the challenges of managing virtual environments. This technology enables enterprises to improve visibility and control with virtual infrastructures.

Overview

Two options are available to install and deploy operands for Infrastructure management.

Option 1. Install Infrastructure management as a containerized deployment (podified)

- To install Infrastructure management as a containerized deployment, see Deploying Infrastructure management as a containerized deployment (podified).

Option 2. Install Infrastructure management as a virtual machine appliance

- To install Infrastructure management as a virtual machine appliance, see Deploying Infrastructure management as a virtual machine appliance.

Option 1. Deploying Infrastructure management as a containerized deployment (podified).

Complete these steps to install Infrastructure management as a containerized deployment.

- Prerequisites
- Deploy the ibm-management-im-install operand

Prerequisites

- Ensure the following operators for **`Infrastructure management`** have been installed.
  - **`ibm-management-im-install`** for Infrastructure management has been installed. For more information, see Infrastructure management.
- Ensure IBM Cloud Pak® is setup for LDAP authentication. You need an existing user-group from this LDAP repository to use for the Infrastructure management deployment.

Deploy the ibm-management-im-install operand

Install an instance of the Infrastructure management using the Red Hat® OpenShift® Container Platform console or the CLI (command-line tools).

1. Create an installation instance CR file **<im-install-cr.yaml>** using the following yaml file and update the **applicationDomain**, **imagePullSecret**, **initialAdminGroupName** and accept license. Add **storageClassName: <storage-Class-Name>** in the create an installation instance CR file **<im-install-cr.yaml>**.

```
apiVersion: infra.management.ibm.com/v1alpha1
kind: IMInstall
metadata:
  labels:
    app.kubernetes.io/instance: ibm-infra-management-install-operator
    app.kubernetes.io/managed-by: ibm-infra-management-install-operator
    app.kubernetes.io/name: ibm-infra-management-install-operator
  name: im-iminstall
  namespace: cp4aiops
spec:
  applicationDomain: <YOUR_IM_HTTPD_ROUTE>
  imagePullPolicy: Always
  imagePullSecret: ibm-entitlement-key
  initialAdminGroupName: <YOUR_LDAP_USER_GROUP>
  storageClassName: <RWX-storage-class-name>
  license:
    accept: true
```

   Where:

   YOUR_IM_HTTPD_ROUTE is a user-defined route, which must include a name for your installation plus part of your IBM Cloud Pak Automation console console's route. Use the following oc command to obtain the console route and modify it to derive your Infrastructure management route.

   ```
   oc get ingress.config.openshift.io -o=jsonpath='{.items[0].spec.domain}'
   ```

   Example output: apps.mycluster.myibm.com

   Add inframgmtinstall to the output to create YOUR_IM_HTTPD_ROUTE

   For example: inframgmtinstall.apps.mycluster.myibm.com

   YOUR_LDAP_USER_GROUP is an existing user-group defined in your LDAP repository. As part of the initial setup, this LDAP group is created in Infrastructure management to match your existing LDAP group by name, and assigned an account role which facilitates SSO login.

   RWX-storage-class-name is a storage class name.

   Important:

   You must specify an LDAP user-group and it must contain at least one user that is able to login to the IBM Cloud Pak Automation console. For example, you have an existing LDAP group that is named group100 and a user with the username user100 is a member of the group. You enter group100 for the value of **<YOUR_LDAP_USER_GROUP>**.

   ACCEPT LICENSE set the accept value to **true** to accept the license.

2. Deploy the installation instance CR yaml using one of the following methods:

   - Using the CLI login to your OpenShift cluster where Infrastructure Automation is installed. For example,

     ```
     oc login --token=`<sha256~EUe-BThLn-qRm32K16QDqLMCQZz5VKlM42JKIj2-U7M>` --
     server=https://`<api.mycluster.myibm.com>`:6443
     ```

     ```
     oc apply -f im-install-cr.yaml
     ```

   - Using the OpenShift Console.

     Create an IMInstall instance to deploy Infrastructure management.

- Navigate to **Operators > Installed Operators.**
- Switch the project to cp4aiops.
- Click IBM Infrastructure Management Install.
- Click Create Instance in the IMInstall tile.
- Switch to YAML view.
- Copy and paste the installation instance yaml created in step 1.
- Click Create.

3. Verify the pods are running (It can take a few minutes for the pods to start) with the command,

   ```
   oc get pods -n cp4aiops
   ```

   The following Infrastructure management pods should be running.

   - 1-event-handle-*
   - 1-generic-*
   - 1-priority-*
   - 1-remote-console-*
   - 1-reporting-*
   - 1-schedule-*
   - 1-ui-*
   - 1-web-service-*
   - httpd-*
   - ibm-infra-management-application-*
   - memcached-*

Option 2. Deploying Infrastructure management as a virtual machine appliance

Complete these steps to install Infrastructure management as a virtual machine appliance.

- Prerequisites
- Step 1. Download the appliance image
- Step 2. Install and configure the appliance
- Step 3. Configure OIDC for Infrastructure management appliance

Prerequisites

- Ensure you enable the operators for **`Infrastructure management`** by opening the installation YAML file. Locate the **`pakModules`** section, and change **`enabled: false`** to **`enabled: true`**. Enable these Infrastructure management-related operators:
  - **`ibm-management-im-install`** for Infrastructure management has been installed. For more information, see Infrastructure management.
- You must configure and connect an LDAP directory with IBM Cloud Pak®. You must have an LDAP group in your configuration for IBM Cloud Pak with users defined who will access Infrastructure management.

Step 1. Download the Infrastructure management appliance package for your environment.

Follow the steps in Install the Infrastructure management appliance for your virtual environment.

Step 2. Install and configure the Infrastructure management appliance.

Follow the steps in Install the Infrastructure management appliance for your virtual environment.

Step 3. Configure OIDC integration between IBM Cloud Pak® and the Infrastructure management appliance.

Follow the steps in Configuring OIDC for Infrastructure management appliance only.

Note:

Only OIDC-based authentication (OpenID Connect) is supported. The configuration of OpenID Connect (OIDC) is required for integration with Infrastructure management and IBM Cloud Pak®.

OIDC Configuration Steps for Infrastructure Management Appliance only

Complete the following steps to configure Infrastructure management virtual machine appliance with OIDC (OpenID Connect).

- Prerequisites
- Step 1. Register Infrastructure management instance with IAM as an OIDC client
- Step 2. Import the Root CA certificate to Infrastructure management
- Step 3. Configure the Administrative UI
- Step 4. Integrating Infrastructure management with IBM Cloud Pak for AIOps

Prerequisites

- Before you configure OIDC make sure you install the Infrastructure management virtual machine appliance. For more information, see Installation and Upgrade.
- Single sign-on with Infrastructure management and IBM Cloud Pak for AIOps requires an LDAP server connection.

  For more information about adding an LDAP connection, see Configuring LDAP connection.

Step 1. Register Infrastructure management instance with IAM as an OIDC client

In order to enable single sign-on (SSO) for Infrastructure management with OIDC, the Infrastructure management instance needs to register as an OIDC client with Identity and Access Management (IAM). Complete these steps on the IBM Cloud Pak for AIOps cluster.

You can register Infrastructure management as an OIDC client with IAM using the **`cloudctl`** command.

The registration method requires the following registration payload in a file "registration.json":

```
{
  "token_endpoint_auth_method":"client_secret_basic",
  "client_id": "<YOUR_CLIENT_ID>",
  "client_secret": "<YOUR_CLIENT_SECRET>",
  "scope":"openid profile email",
  "grant_types":[
     "authorization_code",
     "client_credentials",
     "password",
     "implicit",
     "refresh_token",
     "urn:ietf:params:oauth:grant-type:jwt-bearer"
  ],
  "response_types":[
     "code",
     "token",
     "id_token token"
```

```
  ],
  "application_type":"web",
  "subject_type":"public",
  "post_logout_redirect_uris":[
     "https://<YOUR_CLOUD_PAK_ROUTE>"   ],
  "preauthorized_scope":"openid profile email general",
  "introspect_tokens":true,
  "trusted_uri_prefixes":[
     "https://<YOUR_CLOUD_PAK_ROUTE>/"    ],
  "redirect_uris":
["https://<YOUR_CLOUD_PAK_ROUTE>/auth/liberty/callback","https://<INFRA_MGMT_URL>/oidc_login/r
edirect_uri"]
}
```

Example registration payload (for reference only):

```
{
  "token_endpoint_auth_method":"client_secret_basic",
  "client_id": "AaNzNVFsSjlLVkl6Zk5hZ01MRzJVaVdnbFcxNGl5cnQK",
  "client_secret": "AaNVNzF4ZUxNSVBQUHZHdG1xQmNsTTFOWmNUUGlnYUkK",
  "scope":"openid profile email",
  "grant_types":[
     "authorization_code",
     "client_credentials",
     "password",
     "implicit",
     "refresh_token",
     "urn:ietf:params:oauth:grant-type:jwt-bearer"
  ],
  "response_types":[
     "code",
     "token",
     "id_token token"
  ],
  "application_type":"web",
  "subject_type":"public",
  "post_logout_redirect_uris":["https://cp-console.apps.mycluster.mydomain.com"],
  "preauthorized_scope":"openid profile email general",
  "introspect_tokens":true,
  "trusted_uri_prefixes":["https://cp-console.apps.mycluster.mydomain.com/"],
  "redirect_uris":["https://cp-
console.apps.mycluster.mydomain.com/auth/liberty/callback","https://im-
mycluster.mydomain.com/oidc_login/redirect_uri"]
}
```

1. Create a file named **registration.json** from the example template. Replace the values in the example template payload registration with the actual values based on your installation.

   - **YOUR_CLIENT_ID** Your base64 encoded character string.
   - **YOUR_CLIENT_SECRET** Your base64 encoded character string.

     Note: The **<YOUR_CLIENT_ID>** and **<YOUR_CLIENT_SECRET>** need to be generated. The values can be any string, but normally a 32 character string that is base64 encoded is used. You can use BASE64 to encode your character string. For more information, see: BASE64. Make a note of the values you generate for **<YOUR_CLIENT_ID>** and **<YOUR_CLIENT_SECRET>**. You will use the values in the next section to update the Apache configuration file.

     Example command that uses base64 to encode a character string:

     ```
     #
     # Generate two encrypted streams from some longer-than-32-characters strings
     #
     echo There is a huge white elephant in LA zoo |base64
     echo 12345678901234567890123456789012345 |base64
     ```

   - **YOUR_CLOUD_PAK_ROUTE** The URL of the IBM Cloud Pak for AIOps console.
   - **INFRA_MGMT_URL** The URL of the Infrastructure management virtual machine appliance.
   - **post_logout_redirect_uris** The URL of the IBM Cloud Pak for AIOps console.

- **`trusted_uri_prefixes`** The URL of the IBM Cloud Pak for AIOps console with "forward slash" /.
- **`redirect_uris`** The URL of the IBM Cloud Pak for AIOps console with the path to call back and the URL of the Infrastructure management host with the path to the redirect_uri.

Note: You can run the following command on the IBM Cloud Pak for AIOps cluster to determine the URL of the IBM Cloud Pak for AIOps console:

```
oc get routes cp-console -o=jsonpath='{.spec.host}' -n ibm-common-services
```

2. After the file **`registration.json`** is completed, log in and run the command to register Infrastructure management as an OIDC client.

   Note: Include the **`-n kube-system`** to specify this project, or the **`cloudctl iam`** command can fail.

   ```
   cloudctl login -a https://<YOUR_CLOUD_PAK_ROUTE> -n kube-system
   ```

   Example **`cloudctl iam`** command:

   ```
   cloudctl iam oauth-client-register -f registration.json
   ```

Step 2. Import the Root CA certificate to the Infrastructure management appliance

1. Retrieve the cluster CA cert by running this command on the cluster:

   ```
   oc get secret -n ibm-common-services ibmcloud-cluster-ca-cert -o
   jsonpath='{.data.ca.crt}'| base64 --decode
   ```

   Note: When copying this command there can be some additional characters added that can cause incorrect command syntax. You can copy the first portion **`oc get secret -n ibm-common-services ibmcloud-cluster-ca-cert -o jsonpath=`** and manually add the remaining syntax **`'{.data.ca\.crt}'| base64 --decode`**.

2. Copy and paste the output to a file, for example **`ibm_cp_im.crt`**

3. Edit the **`ibm_cp_im.crt`** file, and change:

   - **`BEGIN CERTIFICATE`** to **`BEGIN TRUSTED CERTIFICATE`**
   - **`END CERTIFICATE`** to **`END TRUSTED CERTIFICATE`**

   Note: The following steps must be completed by logging in to the Infrastructure management appliance system as root user:

4. Copy the updated **`ibm_cp_im.crt`** file to the Infrastructure management appliance and save it in the directory: **`/etc/pki/ca-trust/source/anchors`**

5. Run the command:

   ```
   update-ca-trust
   ```

6. Restart the evm server by running the command:

   ```
   systemctl restart evmserverd
   ```

7. Copy the Apache OIDC template configuration file with these steps:

   ```
   export TEMPLATE_DIR="/opt/IBM/infrastructure-management-appliance/TEMPLATE"
   ```

   ```
   cp ${TEMPLATE_DIR}/etc/httpd/conf.d/manageiq-remote-user-openidc.conf /etc/httpd/conf.d/
   ```

   ```
   cp ${TEMPLATE_DIR}/etc/httpd/conf.d/manageiq-external-auth-openidc.conf.erb
   /etc/httpd/conf.d/manageiq-external-auth-openidc.conf
   ```

8. The Apache **`/etc/httpd/conf.d/manageiq-external-auth-openidc.conf`** configuration file must be updated with installation-specific values. Replace the contents of the file with the actual values based on the installation.

   Sample configuration file:

```
LoadModule          auth_openidc_module modules/mod_auth_openidc.so
ServerName          https://<YOUR_IM_APPLIANCE_HOSTNAME>
LogLevel            warn

OIDCClientID                    <YOUR_CLIENT_ID>
OIDCClientSecret                <YOUR_CLIENT_SECRET>
OIDCRedirectURI
https://<YOUR_IM_APPLIANCE_HOSTNAME>/oidc_login/redirect_uri
OIDCCryptoPassphrase            <PASSPHRASE>
OIDCOAuthRemoteUserClaim        sub
OIDCRemoteUserClaim             name

OIDCOAuthClientID                   <YOUR_CLIENT_ID>
OIDCOAuthClientSecret               <YOUR_CLIENT_SECRET>
OIDCOAuthIntrospectionEndpoint
https://<YOUR_CLOUD_PAK_ROUTE>/idprovider/v1/auth/introspect
OIDCOAuthIntrospectionEndpointAuth client_secret_basic

OIDCProviderIssuer                  https://127.0.0.1:443/idauth/oidc/endpoint/OP
OIDCProviderAuthorizationEndpoint
https://<YOUR_CLOUD_PAK_ROUTE>/idprovider/v1/auth/authorize
OIDCProviderTokenEndpoint
https://<YOUR_CLOUD_PAK_ROUTE>/idprovider/v1/auth/token
OIDCProviderJwksUri                 https://<YOUR_CLOUD_PAK_ROUTE>/idprovider/v1/auth/jwk
OIDCProviderEndSessionEndpoint
https://<YOUR_CLOUD_PAK_ROUTE>/idprovider/v1/auth/logout

OIDCScope                         "openid email profile"
OIDCResponseMode                  "query"
OIDCProviderTokenEndpointAuth     client_secret_post

OIDCPassUserInfoAs json
OIDCSSLValidateServer off
OIDCOAuthSSLValidateServer off
OIDCHTTPTimeoutShort 10

OIDCCacheEncrypt On
<Location /oidc_login>
  AuthType  openid-connect
  Require   valid-user
</Location>

<Location /ui/service/oidc_login>
  AuthType openid-connect
  Require valid-user
  Header set Set-Cookie "miq_oidc_access_token=%{OIDC_access_token}e; Max-Age=10;
Path=/ui/service"
</Location>

<LocationMatch ^/api(?!\/(v[\d\.]+\/)?product_info$)>
  SetEnvIf X-Auth-Token  '^.+$'                 let_api_token_in
  SetEnvIf X-MIQ-Token   '^.+$'                 let_sys_token_in
  SetEnvIf X-CSRF-Token  '^.+$'                 let_csrf_token_in

  AuthType  oauth20
  AuthName  "External Authentication (oidc) for API"

  Require   valid-user
  Order     Allow,Deny
  Allow from env=let_api_token_in
  Allow from env=let_sys_token_in
  Allow from env=let_csrf_token_in
  Satisfy   Any
</LocationMatch>

RequestHeader unset X-REMOTE-USER
RequestHeader unset X-REMOTE_USER
RequestHeader unset X_REMOTE-USER
RequestHeader unset X_REMOTE_USER
RequestHeader set X_REMOTE_USER %{OIDC_CLAIM_PREFERRED_USERNAME}e
env=OIDC_CLAIM_PREFERRED_USERNAME
RequestHeader set X_EXTERNAL_AUTH_ERROR %{EXTERNAL_AUTH_ERROR}e env=EXTERNAL_AUTH_ERROR
```

```
RequestHeader set X_REMOTE_USER_EMAIL %{OIDC_CLAIM_EMAIL}e env=OIDC_CLAIM_EMAIL
RequestHeader set X_REMOTE_USER_FIRSTNAME %{OIDC_CLAIM_GIVEN_NAME}e
env=OIDC_CLAIM_GIVEN_NAME
RequestHeader set X_REMOTE_USER_LASTNAME %{OIDC_CLAIM_FAMILY_NAME}e
env=OIDC_CLAIM_FAMILY_NAME
RequestHeader set X_REMOTE_USER_FULLNAME %{OIDC_CLAIM_NAME}e env=OIDC_CLAIM_NAME
RequestHeader set X_REMOTE_USER_GROUPS %{OIDC_CLAIM_GROUPS}e env=OIDC_CLAIM_GROUPS
RequestHeader set X_REMOTE_USER_DOMAIN %{OIDC_CLAIM_DOMAIN}e env=OIDC_CLAIM_DOMAIN
```

- `YOUR_IM_APPLIANCE_HOSTNAME` Specifies the hostname of the Infrastructure management appliance server.
- `YOUR_CLIENT_ID` The client ID used for registering Infrastructure management as an OIDC client with IAM.
- `YOUR_CLIENT_SECRET` The client secret that is used for registering Infrastructure management as an OIDC client with IAM.
- `YOUR_CLOUD_PAK_ROUTE` The URL of the IBM Cloud Pak UI console.
- `OIDCCryptoPassphrase` Can be any arbitrary alpha-numeric string.

Note: `YOUR_CLIENT_ID` and `YOUR_CLIENT_SECRET` values are generated when you register Infrastructure Management as an OIDC client.

Example configuration file (for reference only):

```
LoadModule          auth_openidc_module modules/mod_auth_openidc.so
ServerName          https://im-mycluster.mydomain.com
LogLevel            warn

OIDCCLientID                        AaNzNVFsSjlLVkl6Zk5hZ01MRzJVaVdnbFcxNGl5cnQK
OIDCClientSecret                    AaNVNzF4ZUxNSVBQUHZHdG1xQmNsTTFOWmNUUGlnYUkK
OIDCRedirectURI                     https://im-
mycluster.mydomain.com/oidc_login/redirect_uri
OIDCCryptoPassphrase                alphabeta
OIDCOAuthRemoteUserClaim            sub
OIDCRemoteUserClaim                 name

OIDCOAuthClientID                   AaNzNVFsSjlLVkl6Zk5hZ01MRzJVaVdnbFcxNGl5cnQK
OIDCOAuthClientSecret
AaNVNzF4ZUxNSVBQUHZHdG1xQmNsTTFOWmNUUGlnYUkK
OIDCOAuthIntrospectionEndpoint      https://cp-
console.apps.mycluster.mydomain.com/idprovider/v1/auth/introspect
OIDCOAuthIntrospectionEndpointAuth client_secret_basic

OIDCProviderIssuer                  https://127.0.0.1:443/idauth/oidc/endpoint/OP
OIDCProviderAuthorizationEndpoint   https://cp-
console.apps.mycluster.mydomain.com/idprovider/v1/auth/authorize
OIDCProviderTokenEndpoint           https://cp-
console.apps.mycluster.mydomain.com/idprovider/v1/auth/token
OIDCProviderJwksUri                 https://cp-
console.apps.mycluster.mydomain.com/idprovider/v1/auth/jwk
OIDCProviderEndSessionEndpoint      https://cp-
console.apps.mycluster.mydomain.com/idprovider/v1/auth/logout

OIDCScope                           "openid email profile"
OIDCResponseMode                    "query"
OIDCProviderTokenEndpointAuth       client_secret_post

OIDCPassUserInfoAs json
OIDCSSLValidateServer off
OIDCOAuthSSLValidateServer off
OIDCHTTPTimeoutShort 10

OIDCCacheEncrypt On
<Location /oidc_login>
  AuthType  openid-connect
  Require   valid-user
</Location>

<Location /ui/service/oidc_login>
  AuthType openid-connect
  Require valid-user
  Header set Set-Cookie "miq_oidc_access_token=%{OIDC_access_token}e; Max-Age=10;
Path=/ui/service"
</Location>
```

```
<LocationMatch ^/api(?!\/(v[\d\.]+\/)?product_info$)>
  SetEnvIf X-Auth-Token   '^.+$'                 let_api_token_in
  SetEnvIf X-MIQ-Token    '^.+$'                 let_sys_token_in
  SetEnvIf X-CSRF-Token   '^.+$'                 let_csrf_token_in

  AuthType  oauth20
  AuthName  "External Authentication (oidc) for API"

  Require   valid-user
  Order     Allow,Deny
  Allow from env=let_api_token_in
  Allow from env=let_sys_token_in
  Allow from env=let_csrf_token_in
  Satisfy   Any
</LocationMatch>

RequestHeader unset X-REMOTE-USER
RequestHeader unset X-REMOTE_USER
RequestHeader unset X_REMOTE-USER
RequestHeader unset X_REMOTE_USER
RequestHeader set X_REMOTE_USER %{OIDC_CLAIM_PREFERRED_USERNAME}e
env=OIDC_CLAIM_PREFERRED_USERNAME
RequestHeader set X_EXTERNAL_AUTH_ERROR %{EXTERNAL_AUTH_ERROR}e env=EXTERNAL_AUTH_ERROR
RequestHeader set X_REMOTE_USER_EMAIL %{OIDC_CLAIM_EMAIL}e env=OIDC_CLAIM_EMAIL
RequestHeader set X_REMOTE_USER_FIRSTNAME %{OIDC_CLAIM_GIVEN_NAME}e
env=OIDC_CLAIM_GIVEN_NAME
RequestHeader set X_REMOTE_USER_LASTNAME %{OIDC_CLAIM_FAMILY_NAME}e
env=OIDC_CLAIM_FAMILY_NAME
RequestHeader set X_REMOTE_USER_FULLNAME %{OIDC_CLAIM_NAME}e env=OIDC_CLAIM_NAME
RequestHeader set X_REMOTE_USER_GROUPS %{OIDC_CLAIM_GROUPS}e env=OIDC_CLAIM_GROUPS
RequestHeader set X_REMOTE_USER_DOMAIN %{OIDC_CLAIM_DOMAIN}e env=OIDC_CLAIM_DOMAIN
```

9. Restart Apache on the appliance.

```
systemctl restart httpd
```

Step 3. Configure the Administrative UI

Update the Appliance Administrative UI to be OIDC aware and function. Complete these steps on each UI-enabled Infrastructure management appliance.

1. Log in as `admin`.
2. Select the **Settings** > **Application Settings**, select **"Server "EVM [1]" (current)"** under Zones, then select the **Authentication** tab.
3. In the **Authentication** section, set the **Mode** to `External (httpd)`
4. In the **External Authentication (HTTPd) Settings** section, set **Provider Type** to `Enable OpenID-Connect`.
   - **Note:** This setting enables the OIDC login button on the login screen that redirects to the OIDC protected page for authentication, and supports the OIDC logout process.
5. Optional: In the **External Authentication (HTTPd) Settings** section, select **Enable Single Sign-On**.
   - **Note:** If you select this option, the initial access to the Appliance Administrative UI will redirect to the OIDC Identity Provider authentication screen.
6. In the **Role Settings** section, select the **Get User Groups from External Authentication (HTTPd)** setting.
7. Click Save.
8. Select **Access Control**, click **Groups** and make sure the user's groups are created on the Appliance and appropriate roles are assigned to those groups. The user's groups to be added in Infrastructure management should have the same names as the groups defined in the LDAP server that is configured in the IBM Cloud Pak console.
   1. Under the **Access Control**, click **Groups**.
   2. Click **Configuration**, then **Add a new Group**.
   3. Enter your existing LDAP group name in the **Description** field. For example, `im_ldap_group`.

4. Select the **Role** for this LDAP group. For example, select the **EvmRole-super_administrator** to map to this group. The Infrastructure management administrator can map to the roles that make sense for the user group.
5. Select **My Company** for **Project/Tenant**.
6. Click Add.

Note: Access control in Infrastructure management is based on group membership as roles are assigned to groups. When Infrastructure management is integrated with IBM Cloud Pak for AIOps using single sign-on (SSO), it looks at the user's group membership in the identity token and checks if that group exists in Infrastructure management. If the group doesn't exist, then access is denied. You must create at least one group with the role **EvmRole-super_administrator** and assign one user to this group. This user will be the Admin user for Infrastructure management and referred to as **`IM_ADMIN_LDAP_USER`**. The **`IM_ADMIN_LDAP_USER`** must be a member of the LDAP group with the role **EvmRole-super_administrator** in Infrastructure management.

Important: You must create the groups in Infrastructure management that match your existing LDAP groups by name, and assign the groups account roles. At least one group to which the user belongs in LDAP that IBM Cloud Pak for AIOps is configured to use must also be created in Infrastructure management. You must assign a proper role to this group in Infrastructure management.

Example: In LDAP a group that is named **`imgroup100`** exists and a user with username **`imuser100`** is a member of the group. The user **`imuser100`** and the group **`imgroup100`** must be created in Infrastructure management.

Step 4. Integrating Infrastructure management with IBM Cloud Pak for AIOps

Enable navigation to Infrastructure management within the IBM Cloud Pak console.

Complete the following steps on a Linux system. You can use the boot node from the HUB cluster where IBM Cloud Pak for AIOps is installed. These steps enable navigation to Infrastructure management from the IBM Cloud Pak console:

1. Clone the IBM Cloud Pak for AIOps samples GitHub repository to obtain the menu customization script. You must run the script on a Linux operating system.

   Download the supporting subdirectories to make the script run successfully. The files and subdirectories that are required to run the script are

   ```
   im-appliance-link-install.sh
   nav/automate.yml
   nav_template/automate.sh
   ```

2. Install and authenticate **`kubectl`**. For more information, see Installing the Kubernetes CLI (kubectl).

3. Navigate to the directoy **`cp4waiops-samples/infrastructure-automation/infrastructure-management/appliance`**. Set the file permissions on the script and run **`im-appliance-link-install.sh`** to enable navigation to your Infrastructure management instance:

   ```
   chmod 755 ./im-appliance-link-install.sh
   ```

   ```
   ./im-appliance-link-install.sh -i <Infrastructure management URL>
   ```

   - `-i` Is a required parameter that refers to the URL for the Infrastructure management console. For example, `./im-appliance-link-install.sh -i https://vm17-im-mycluster.mydomain.com/`

4. Verify that the Infrastructure management instance is in the IBM Cloud Pak console navigation menu. From the IBM Cloud Pak navigation menu, click **Automate infrastructure** > **Infrastructure management**.

Infrastructure management is integrated with the IBM Cloud Pak console.

Upgrading

Upgrade a deployment of IBM Cloud Pak® for AIOps or Infrastructure Automation.

- Upgrading IBM Cloud Pak for AIOps

- Upgrading Infrastructure Automation

Upgrading IBM Cloud Pak for AIOps

Upgrade an existing online or offline deployment of IBM Cloud Pak® for AIOps 4.1.0 or later to 4.2.0.

You cannot use these instructions to upgrade deployments of IBM Cloud Pak for AIOps 3.7.2 or earlier.

Upgrade paths

IBM Cloud Pak for AIOps uses a versioning system of X.Y.Z, where X=version, Y=release, and Z=patch. For example, IBM Cloud Pak for AIOps 4.2.1 is version 4, release 2, patch 1. IBM Cloud Pak for AIOps upgrades can be release upgrades or patch upgrades.

You can upgrade by a maximum of one release at a time. If the release that you want to upgrade from is more than one release behind the release that you are upgrading to, then you must upgrade sequentially. For example, you cannot upgrade directly from IBM Cloud Pak for AIOps 3.6.0 to v4.2.0, you must use the following sequence:

1. Use the v3.7.2 documentation to upgrade from IBM Cloud Pak for AIOps 3.6.0 to v3.7.2.
2. Use the v4.1.2 documentation to upgrade from IBM Cloud Pak for AIOps 3.7.2 to v4.1.2.
3. Use the v4.2.0 documentation to upgrade from IBM Cloud Pak for AIOps 4.1.2 to v4.2.0.

Controlling upgrade

Patch upgrades

Online deployments

Upgrades of online IBM Cloud Pak for AIOps deployments to a higher patch, for example from IBM Cloud Pak for AIOps 4.1.0 to 4.1.1, complete automatically if catalog polling is enabled. You elect to enable or disable catalog polling when you install IBM Cloud Pak for AIOps, and you can change this setting after installation if necessary. Automatic patch upgrade can also occur if you have not fixed the **`ibm-operator-catalog`** CatalogSource to a specific image and the **`ibm-operator-catalog`** pods are restarted, even if catalog polling is disabled.

If you do not want your IBM Cloud Pak for AIOps deployment to upgrade automatically when newer patches are available, then you must disable catalog polling and pin the image to a specific version. For more information, see Configuring automatic patch upgrades.

Offline deployments

Upgrades of offline IBM Cloud Pak for AIOps deployments to a higher patch do not occur automatically, as the updated installation images must be manually mirrored to the offline repository first.

Release upgrades

Online deployments

Upgrades of online IBM Cloud Pak for AIOps deployments to a higher release, for example from IBM Cloud Pak for AIOps 3.7.2 to 4.1.0 or from 4.1.0 to 4.2.0 do not complete automatically, even if catalog polling is enabled. For a release upgrade, catalog polling must be enabled and the channel subscription must also be manually edited.

Offline deployments

Upgrades of offline deployments to a higher release version do not occur automatically, as the updated installation images must be manually mirrored to the offline repository first.

Guidance for upgrades that require an Red Hat OpenShift upgrade

Before you upgrade IBM Cloud Pak for AIOps, ensure that you are on a version of Red Hat® OpenShift® Container Platform that your current and target versions of IBM Cloud Pak for AIOps both support. This helps to ensure compatibility and avoid potential issues caused by platform API changes.

If you already have a version of Red Hat OpenShift that your current and target versions of IBM Cloud Pak for AIOps both support but you want to upgrade Red Hat OpenShift, then complete the IBM Cloud Pak for AIOps upgrade before you upgrade Red Hat OpenShift.

Example upgrade scenario

You have IBM Cloud Pak for AIOps 4.1.0 deployed on Red Hat OpenShift 4.10, and you want to upgrade to IBM Cloud Pak for AIOps 4.2.1 on Red Hat OpenShift 4.12.

The following table shows the supported Red Hat OpenShift versions for your current and target versions of IBM Cloud Pak for AIOps:

Table 1. Supported Red Hat OpenShift versions for current and target application versions

IBM Cloud Pak for AIOps version	Supported Red Hat OpenShift versions
4.1.0	4.10, 4.12
4.2.1	4.12

The upgrade steps for this example upgrade scenario are:

1. Upgrade Red Hat OpenShift 4.10 to v4.12.
2. Upgrade IBM Cloud Pak for AIOps 4.1.0 to v4.2.1.

Upgrade procedure

The upgrade procedure depends on whether your deployment is an online or offline (air-gapped) deployment. For more information about online and offline deployments, see Installing IBM Cloud Pak for AIOps.

Online deployments

You can upgrade an online deployment of IBM Cloud Pak for AIOps with the Red Hat OpenShift console or command-line interface (CLI).

- Online upgrade of IBM Cloud Pak for AIOps (console method)
- Online upgrade of IBM Cloud Pak for AIOps (CLI method)

Offline deployments

- Offline upgrade of IBM Cloud Pak for AIOps

Online upgrade of IBM Cloud Pak for AIOps (console method)

Use these instructions to upgrade IBM Cloud Pak® for AIOps 4.1.0 or later to 4.2.0.

This procedure can be used on an online deployment of IBM Cloud Pak for AIOps 4.1.0 or later, and can still be used if the deployment has had hotfixes applied. If you have an offline deployment, follow the instructions in Upgrading IBM Cloud Pak for AIOps (offline).

Before you begin

Notes:

- Ensure that you are on a version of Red Hat OpenShift that your current and target versions of IBM Cloud Pak for AIOps both support. If you already have a qualifying version of Red Hat OpenShift but you want to upgrade it, then complete

the IBM Cloud Pak for AIOps upgrade first. For more information, see Guidance for upgrades that require an Red Hat OpenShift upgrade.

- Some steps must still be performed with the Red Hat® OpenShift® Container Platform command line interface (CLI). Ensure that you are logged in to your Red Hat OpenShift Container Platform cluster with `oc login` for any steps that use the OpenShift command-line interface (CLI).
- Red Hat OpenShift Container Platform requires a user with `cluster-admin` privileges for the following operations:
  - Configure automatic catalog polling
  - Update foundational services
  - Update the operator subscription

Warnings:

- Custom patches, labels, and manual adjustments to IBM Cloud Pak for AIOps resources are lost when IBM Cloud Pak for AIOps is upgraded, and must be manually reapplied after upgrade. For more information, see Manual adjustments are not persisted.
- If you previously increased the size of a PVC directly, then you must follow the correct procedure that is supplied in Adjusting PVCs to ensure that the size is updated by the operator. Failure to do so before upgrading IBM Cloud Pak for AIOps causes the operator to attempt to restore a lower default value for the PVC, and causes an error in your IBM Cloud Pak for AIOps deployment.

Restrictions:

- You cannot use these instructions to upgrade deployments of IBM Cloud Pak for AIOps 3.7.2 or earlier. For more information, see Upgrade paths.
- The upgrade cannot be removed or rolled back.

Upgrade procedure

Follow these steps to upgrade your online IBM Cloud Pak for AIOps deployment.

1. Ensure cluster readiness
2. Configure automatic catalog polling
3. Update foundational services
4. Scale down the AIOpsEdge deployment
5. Update the operator subscription
6. Verify the deployment
7. Post upgrade actions

1. Ensure cluster readiness

Recommended: Take a backup before upgrading. For more information, see Backup and restore.

1. Ensure that your cluster still meets all of the prerequisites for deployment. For more information, see Planning.

   Note: IBM Cloud Pak for AIOps requires that Red Hat OpenShift Container Platform must be version 4.12.

2. Run the IBM Cloud Pak for AIOps prerequisite checker script.

   The prerequisite checker script ensures that your Red Hat OpenShift Container Platform cluster is correctly set up for an IBM Cloud Pak for AIOps upgrade. When you run the prerequisite checker script, you must run the script in the same project (namespace) that IBM Cloud Pak for AIOps is installed in.

   For more information about the script, including how to download and run it, see github.com/IBM.

   Important: The prerequisite checker script might show inadequate resources in the *Resource Summary* because the script does not account for resources already being in use by the upgrading deployment. This can be ignored, as can the following message: *[FAIL] Small or Large Profile Install Resources*.

   Example output:

```
# ./prereq.sh
Using project "cp4waiops" on server "https://myserver.mycluster.mydomain:6443".
```

```
Starting IBM Cloud Pak for AIOps prerequisite checker v4.2...

[INFO] ================================Openshift Container Platform Version
Check================================
[INFO] Checking OCP Version. Compatible Versions of OCP are v4.12.x
[INFO] OCP version 4.12.18 is compaitble
[INFO] ================================Openshift Container Platform Version
Check================================

[INFO] ================================Entitlement Pull
Secret================================
[INFO] Checking whether the Entitlement secret or Global pull secret is configured
correctly.
[INFO] Checking if the job 'cp4waiops-entitlement-key-test-job' already exists.
[INFO] The job with name 'cp4waiops-entitlement-key-test-job' was not found, so moving
ahead and creating it.
[INFO] Creating the job 'cp4waiops-entitlement-key-test-job'
job.batch/cp4waiops-entitlement-key-test-job created
[INFO] Verifying if the job 'cp4waiops-entitlement-key-test-job' completed successfully..
[INFO] SUCCESS! Entitlement secret is configured correctly.
job.batch "cp4waiops-entitlement-key-test-job" deleted
[INFO] ================================Entitlement Pull
Secret================================

[INFO] ================================Storage Provider================================
[INFO] Checking storage providers
[INFO] No IBM Storage Fusion Found... Skipping configuration check.

[INFO] No Portworx StorageClusters found with "Running" or "Online" status. Skipping
configuration check for Portworx.
[INFO] Openshift Data Foundation found.
[INFO] No IBM Cloud Storage found... Skipping configuration check for IBM Cloud Storage
Check.

Checking Openshift Data Foundation Configuration...
Verifying if Red Hat Openshift Data Foundation pods are in "Running" or "Completed"
status
[INFO] Pods in openshift-storage project are "Running" or "Completed"
[INFO] ocs-storagecluster-ceph-rbd exists.
[INFO] ocs-storagecluster-cephfs exists.
[INFO] No warnings or failures found when checking for Storage Providers.
[INFO] ================================Storage Provider================================

[INFO] ================================Small or Large Profile Install
Resources================================
[INFO] Checking for cluster resources

[INFO] ================================Resource
Summary============================================================
[INFO]                                        Nodes   |      vCPU      |     Memory(GB)
[INFO] Small profile(available/required)  [  14 / 3 ]   [  255 / 62 ]       [  479 / 140
]
[INFO] Large profile(available/required)  [  14 / 10 ]  [  255 / 162 ]      [  479 / 372
]
[INFO] ================================Resource
Summary=======================================
[INFO] Cluster currently has resources available to create a large profile of Cloud Pak
for AIOps
[INFO] ================================Small or Large Profile Install
Resources=================


[INFO] ================================Prerequisite Checker Tool
Summary================================
   [  PASS  ] Openshift Container Platform Version Check
   [  PASS  ] Entitlement Pull Secret
   [  PASS  ] Storage Provider
   [  PASS  ] Small or Large Profile Install Resources
[INFO] ================================Prerequisite Checker Tool
Summary================================
```

3. Delete any evicted **connector-orchestrator** pods.

    1. From the **Project** list, select the project (namespace) that IBM Cloud Pak for AIOps is deployed in if your deployment is namespace scoped, or **openshift-operators** if your deployment has a cluster-wide scope.

    2. From the left menu, select **Workloads** > **Pods**, and search for **connector-orchestrator**.

    3. If there are any **connector-orchestrator** pods with a **STATUS** of **Evicted**, then select **Delete Pod** from the three dots menu at the end of the row.

2. Configure automatic catalog polling

Ensure that your catalog is set to automatically poll for the latest images.

Your **ibm-operator-catalog** CatalogSource object can be configured to automatically poll for the latest catalog version, and to retrieve it if one is available. Polling for updates is enabled by configuring the polling attribute, **spec.updateStrategy.registryPoll**.

You might have already elected to automatically accept updates by adding the polling attribute to your **ibm-operator-catalog** YAML when you installed IBM Cloud Pak for AIOps, installed an IBM Cloud Pak for AIOps hotfix from IBM support, or when you installed another IBM Cloud Pak®.

Use the following steps to check whether you already have a polling attribute set, and to configure it if you do not.

Note: **ibm-operator-catalog** also contains the catalogs for other IBM Cloud Paks. If you have multiple IBM Cloud Paks installed on your cluster and you enable the polling attribute, then automatic update is configured for all of them.

1. Log in to your OpenShift cluster's console.

2. Go to **Administration** > **Cluster Settings**.

3. Click **Configuration**, and then scroll down and click **OperatorHub**.

4. Select **Sources**, and then scroll down and click **ibm-operator-catalog**.

5. Click **YAML** to switch to the YAML view.

6. If there is not a **spec.updateStrategy** section, or **spec.image** is not set to **icr.io/cpopen/ibm-operator-catalog:latest**, then update the YAML to have the following in the **spec** block, and save it.

```
spec:
  displayName: ibm-operator-catalog
  publisher: IBM Content
  sourceType: grpc
  image: icr.io/cpopen/ibm-operator-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
```

3. Update foundational services

IBM Cloud Pak® foundational services, which is part of your IBM Cloud Pak for AIOps deployment, must be at version 3.23 or higher before you upgrade IBM Cloud Pak for AIOps.

Use the following steps to verify that your **ibm-common-service-operator** subscription is set to version 3.23 or higher, and to set it to a qualifying version if it is not.

1. Run the following command to find out what version of foundational services you have installed.

```
oc get csv -A | grep ibm-common-service-operator
```

    If the version returned is v3.23 or higher, then you do not need to update foundational services and you must skip the rest of this section and proceed to step 4, Scale down the AIOpsEdge deployment.

2. Download the Common Services upgrade script, **upgrade_common_services.sh**, from github.com/IBM.

3. Run the following command from the directory that you downloaded the Common Services upgrade script to. This script must be run by a user with **cluster-admin** privilege.

   ```
   ./cp4waiops-samples/upgrade/upgrade_common_services.sh -a -c v3.23
   ```

   Important: You must only run this script if your version of foundational services is less than v3.23.

4. When **upgrade_common_services.sh** completes, run the following commands to verify that the **ibm-common-service-operator** channel is set to version 3.23 or higher in the subscription and in the ClusterServiceVersion (CSV) before you continue.

   1. Check the subscription.

      ```
      oc get subscription ibm-common-service-operator -n ibm-common-services -o
      jsonpath='{.spec.channel}'
      ```

      Example output:

      ```
      oc get subscription ibm-common-service-operator -n ibm-common-services -o
      jsonpath='{.spec.channel}'
      'v3.23'
      ```

   2. Check the CSV.

      ```
      oc get csv -A | grep ibm-common-service-operator
      ```

      Example output:

      ```
      oc get csv -A | grep ibm-common-service-operator
      ibm-common-services   ibm-common-service-operator.v3.23.7   IBM Cloud Pak
      foundational services   3.23.7   Succeeded
      cp4aiops              ibm-common-service-operator.v3.23.7   IBM Cloud Pak
      foundational services   3.23.7   Succeeded
      ```

5. The foundational services upgrade commences, and will take approximately 30 - 60 minutes.

   You can run the following command to check the status of **ZenService**. When the foundational services upgrade is complete, this command will have a *STATUS* of **Completed**. Do not proceed until the upgrade has completed.

   ```
   oc get zenservice -A -o custom-
   columns='KIND:.kind,NAME:.metadata.name,NAMESPACE:.metadata.namespace,VERSION:status.curr
   entVersion,STATUS:.status.zenStatus,PROGRESS:.status.Progress,MESSAGE:.status.ProgressMes
   sage'
   ```

   Example output from a successful foundational services upgrade:

   ```
   KIND         NAME                  NAMESPACE   VERSION   STATUS      PROGRESS   MESSAGE
   ZenService   iaf-zen-cpdservice    cp4waiops    4.8.0     Completed   100%       The
   Current Operation Is Completed
   ```

4. Scale down the AIOpsEdge deployment

1. Log in to your OpenShift cluster's console.
2. Go to **Workloads** > **Deployments**.
3. From the **Project** list, select the project (namespace) that IBM Cloud Pak for AIOps is deployed in.
4. Enter **aiopsedge-operator-controller-manager** in the **Name** search field. The **aiopsedge-operator-controller-manager deployment** deployment is listed.
5. Click on the **aiopsedge-operator-controller-manager** deployment. The deployment details for **aiopsedge-operator-controller-manager** are displayed, with a number of pods graphic shown first.

6. Click the down arrow next to the pods graphic until there are 0 pods.
7. Do not proceed until the pod graphic displays **`Scaled to 0`**. This takes a few seconds.

5. Update the operator subscription

Use the following steps to update the **`spec.channel`** value of the IBM Cloud Pak for AIOps subscription to the release that you want to upgrade to, v4.2.

1. Log in to your OpenShift cluster's console.
2. Select **Home** > **Search**.
3. From the **Project** list, select the project (namespace) that your IBM Cloud Pak for AIOps subscription is deployed in. This is your IBM Cloud Pak for AIOps project if your deployment is namespace scoped, or **`openshift-operators`** if your deployment has a cluster wide scope.
4. In the **Resources** list, select **`SUB Subscription`**. A list of subscriptions is displayed.
5. Click the subscription that has a **Name** of **`ibm-aiops-orchestrator`**. A new window with the subscription details for **`ibm-aiops-orchestrator`** is displayed.
6. Click the value in the **Update channel** box. A new window called **Change Subscription update channel** is displayed.
7. Change the channel to **`v4.2`** and click **Save**

6. Verify the deployment

Use the following procedure to check the status of your upgraded IBM Cloud Pak for AIOps deployment. It takes around 60-90 minutes for the upgrade to complete (subject to the speed with which images can be pulled).

1. Log in to your OpenShift cluster's console.
2. Click **Operators** > **Installed Operators**.
3. From the **Project** list, select the project (namespace) that IBM Cloud Pak for AIOps is deployed in if your deployment is namespace scoped, or **`openshift-operators`** if your deployment has a cluster-wide scope.
4. Locate **IBM Cloud Pak for AIOps** in the list, and verify that the annotation underneath it shows **`4.2.0`**.
5. Select **IBM Cloud Pak for AIOps** and then click the **IBM Cloud Pak for AIOps** tab.
6. Under **Installations**, look for the entry with the name that you specified for your IBM Cloud Pak for AIOps instance, and verify that it has a **Status** of *Phase: Running*, which means that your deployment is complete and successful.

   (Optional): If you want to see more detail about the status of your deployment's components, select the entry with the name that you specified for your IBM Cloud Pak for AIOps instance, and then switch to the **YAML** view. Scroll down to the **Status** section near the end of the YAML. A component's installation is complete and successful when the component has a value of **`Ready`**.

   Example YAML:

```
status:
  size: small
  customProfileConfigmap: aiops-custom-size-profile
  customProfileValidationStatus: >-
    Custom profile configmap not found, continue installation process without
    customization
  storageclasslargeblock: rook-ceph-rbd
  componentstatus:
    issueresolutioncore: Ready
    kafka: Ready
    aiopsanalyticsorchestrator: Ready
    aiopsedge: Ready
    tunnel: Ready
    lifecycleservice: Ready
```

```
    zenservice: Ready
    flinkcluster: Ready
    cluster: Ready
    elasticsearch: Ready
    kong: Ready
    aiopsui: Ready
    redissentinel: Ready
    <...>
```

(Optional) You can also download and run a status checker script to see information about the status of your deployment. For more information about how to download and run the script, see github.com/IBM.

If the upgrade fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade and Known Issues to help you identify any installation problems.

7. Go to **Administration** > **CustomResourceDefinitions**.
8. Scroll down and select **LifecycleTrigger**, and then **Instances**.
9. Click the returned *aiops* instance, and then click **YAML** to switch to the YAML view.

   Check that the LifecycleTrigger job is complete. Verify that the YAML contains `reason: Ready` and `status: "True"` before continuing to the next step.

   Example output:

```
status:
  conditions:
  - lastTransitionTime: "2024-03-21T03:04:53Z"
    message: Job '84086bada206a5b4b31cc297e3206835' in state 'RUNNING'
    observedGeneration: 2
    reason: Ready
    status: "True"
    type: LifecycleTriggerReady
```

7. Post upgrade actions

1. If you previously setup backup or restore on your deployment, then you must follow the instructions in Upgrading IBM Cloud Pak for AIOps backup and restore artifacts.
2. If the *EXPIRY_SECONDS* environment variable was set for configuring log anomaly alerts, the environment variable was not retained in the upgrade. After the upgrade is completed, set the environment variable again. For more information about setting the variable, see Configuring expiry time for log anomaly alerts.
3. If the `Access Control` page displays custom roles with deprecated permissions after upgrade, see Custom roles with deprecated permissions after upgrade.
4. If you have Netcool or Metrics connections configured, then you must follow the instructions in After upgrade, Netcool and Metrics connections have a red status in the Connections UI.
5. In a proactive ChatOps channel, if you click the Change request ticket URL, but details don't display in the ServiceNow instance, edit the information for the `x_ibm_waiops.admin` user. For more information, see IBM Change Risk Assessment tab in ServiceNow not displaying change risk assessment details.

Online upgrade of IBM Cloud Pak for AIOps (CLI method)

Use these instructions to upgrade IBM Cloud Pak® for AIOps 4.1.0 or later to 4.2.0.

This procedure can be used on an online deployment of IBM Cloud Pak for AIOps 4.1.0 or later, and can still be used if the deployment has had hotfixes applied. If you have an offline deployment, follow the instructions in Upgrading IBM Cloud Pak for

AIOps (offline).

Before you begin

Notes:

- Ensure that you are on a version of Red Hat OpenShift that your current and target versions of IBM Cloud Pak for AIOps both support. If you already have a qualifying version of Red Hat OpenShift but you want to upgrade it, then complete the IBM Cloud Pak for AIOps upgrade first. For more information, see Guidance for upgrades that require an Red Hat OpenShift upgrade.
- Ensure that you are logged in to your Red Hat® OpenShift® Container Platform cluster with `oc login` for any steps that use the OpenShift command-line interface (CLI).
- Red Hat OpenShift Container Platform requires a user with `cluster-admin` privileges for the following operations:
  - Configure automatic catalog polling
  - Update foundational services
  - Update the operator subscription

Warnings:

- Custom patches, labels, and manual adjustments to IBM Cloud Pak for AIOps resources are lost when IBM Cloud Pak for AIOps is upgraded, and must be manually reapplied after upgrade. For more information, see Manual adjustments are not persisted.
- If you previously increased the size of a PVC directly, then you must follow the correct procedure that is supplied in Adjusting PVCs to ensure that the size is updated by the operator. Failure to do so before upgrading IBM Cloud Pak for AIOps causes the operator to attempt to restore a lower default value for the PVC, and causes an error in your IBM Cloud Pak for AIOps deployment.

Restrictions:

- You cannot use these instructions to upgrade deployments of IBM Cloud Pak for AIOps 3.7.2 or earlier. For more information, see Upgrade paths.
- The upgrade cannot be removed or rolled back.

Upgrade procedure

Follow these steps to upgrade your online IBM Cloud Pak for AIOps deployment.

1. Ensure cluster readiness
2. Configure automatic catalog polling
3. Update foundational services
4. Scale down the AIOpsEdge deployment
5. Update the operator subscription
6. Verify the deployment
7. Post upgrade actions

1. Ensure cluster readiness

Recommended: Take a backup before upgrading. For more information, see Backup and restore.

1. Ensure that your cluster still meets all of the prerequisites for deployment. For more information, see Planning.

   Note: IBM Cloud Pak for AIOps requires that Red Hat OpenShift Container Platform must be version 4.12.x.

2. Run the IBM Cloud Pak for AIOps prerequisite checker script.

   The prerequisite checker script ensures that your Red Hat OpenShift Container Platform cluster is correctly set up for an IBM Cloud Pak for AIOps upgrade. When you run the prerequisite checker script, you must run the script in the same project (namespace) that IBM Cloud Pak for AIOps is installed in.

   For more information about the script, including how to download and run it, see github.com/IBM.

Important: The prerequisite checker script might show inadequate resources in the *Resource Summary* because the script does not account for resources already being in use by the upgrading deployment. This can be ignored, as can the following message: *[FAIL] Small or Large Profile Install Resources*.

Example output:

```
# ./prereq.sh
Using project "cp4waiops" on server "https://myserver.mycluster.mydomain:6443".

Starting IBM Cloud Pak for AIOps AI Manager prerequisite checker v4.2...

[INFO] ================================Openshift Container Platform Version
Check================================
[INFO] Checking OCP Version. Compatible Versions of OCP are v4.12.x
[INFO] OCP version 4.12.18 is compaitble
[INFO] ================================Openshift Container Platform Version
Check================================

[INFO] ================================Entitlement Pull
Secret================================
[INFO] Checking whether the Entitlement secret or Global pull secret is configured
correctly.
[INFO] Checking if the job 'cp4waiops-entitlement-key-test-job' already exists.
[INFO] The job with name 'cp4waiops-entitlement-key-test-job' was not found, so moving
ahead and creating it.
[INFO] Creating the job 'cp4waiops-entitlement-key-test-job'
job.batch/cp4waiops-entitlement-key-test-job created
[INFO] Verifying if the job 'cp4waiops-entitlement-key-test-job' completed successfully..
[INFO] SUCCESS! Entitlement secret is configured correctly.
job.batch "cp4waiops-entitlement-key-test-job" deleted
[INFO] ================================Entitlement Pull
Secret================================

[INFO] ================================Storage Provider================================
[INFO] Checking storage providers
[INFO] No IBM Storage Fusion Found... Skipping configuration check.

[INFO] No Portworx StorageClusters found with "Running" or "Online" status. Skipping
configuration check for Portworx.
[INFO] Openshift Data Foundation found.
[INFO] No IBM Cloud Storage found... Skipping configuration check for IBM Cloud Storage
Check.

Checking Openshift Data Foundation Configuration...
Verifying if Red Hat Openshift Data Foundation pods are in "Running" or "Completed"
status
[INFO] Pods in openshift-storage project are "Running" or "Completed"
[INFO] ocs-storagecluster-ceph-rbd exists.
[INFO] ocs-storagecluster-cephfs exists.
[INFO] No warnings or failures found when checking for Storage Providers.
[INFO] ================================Storage Provider================================

[INFO] ================================Small or Large Profile Install
Resources================================
[INFO] Checking for cluster resources

[INFO] ================================Resource
Summary=========================================================
[INFO]                                      Nodes   |      vCPU      |     Memory(GB)
[INFO] Small profile(available/required)  [  14 / 3 ]   [  255 / 62 ]       [  479 / 140
]
[INFO] Large profile(available/required)  [  14 / 10 ]  [  255 / 162 ]      [  479 / 372
]
[INFO] ================================Resource
Summary=============================================
[INFO] Cluster currently has resources available to create a large profile of Cloud Pak
for AIOps
[INFO] ================================Small or Large Profile Install
Resources========================


[INFO] ================================Prerequisite Checker Tool
```

```
Summary=================================
      [  PASS  ] Openshift Container Platform Version Check
      [  PASS  ] Entitlement Pull Secret
      [  PASS  ] Storage Provider
      [  PASS  ] Small or Large Profile Install Resources
[INFO] ================================Prerequisite Checker Tool
Summary=================================
```

3. Set the environment variables that are needed for upgrade.

   If you still have **waiops_var.sh** from when you installed IBM Cloud Pak for AIOps, then run the following steps from the directory that the script is in:

   1. Change both occurrences of **PROJECT_CP4WAIOPS** in **waiops_var.sh** to be **PROJECT_CP4AIOPS**. Save your changes.

   2. Run the following command to source your script and set the environment variables.

      ```
      . ./waiops_var.sh
      ```

   If you do not have **waiops_var.sh** from when you installed IBM Cloud Pak for AIOps, then run the following commands:

   ```
   export PROJECT_CP4AIOPS=<project>
   export INSTALL_MODE_NAMESPACE=<install_namespace>
   ```

   Where

   - **<project>** is the namespace (project) that your IBM Cloud Pak for AIOps subscription is deployed in.
   - **<install_namespace>** is ${PROJECT_CP4AIOPS} if your deployment is namespace scoped, or **openshift-operators** if your deployment has a cluster-wide scope.

4. Delete any evicted **connector-orchestrator** pods.

   1. Run the following command to check if there are any evicted **connector-orchestrator** pods.

      ```
      oc get pods -n ${PROJECT_CP4AIOPS} | grep connector-orchestrator
      ```

   2. Cleanup any evicted connector-orchestrator pods.

      If the previous command returned any pods with a *STATUS* of **Evicted**, then run the following command to delete each of them.

      ```
      oc delete pod -n ${PROJECT_CP4AIOPS} <connector_orchestrator>
      ```

      Where **<connector_orchestrator>** is a pod returned in the previous step.

2. Configure automatic catalog polling

Ensure that your catalog is set to automatically poll for the latest images.

Your **ibm-operator-catalog** CatalogSource object can be configured to automatically poll for the latest catalog version, and to retrieve it if one is available. Polling for updates is enabled by configuring the polling attribute, **spec.updateStrategy.registryPoll**.

You might have already elected to automatically accept updates by adding the polling attribute to your **ibm-operator-catalog** YAML when you installed IBM Cloud Pak for AIOps, installed an IBM Cloud Pak for AIOps hotfix from IBM support, or when you installed another IBM Cloud Pak®.

Use the following steps to check whether you already have a polling attribute set, and to configure it if you do not.

Note: **ibm-operator-catalog** also contains the catalogs for other IBM Cloud Paks. If you have multiple IBM Cloud Paks installed on your cluster and you enable the polling attribute, then automatic update is configured for all of them.

1. Run the following command to view and edit your **ibm-operator-catalog** CatalogSource instance.

```
oc edit catalogsource ibm-operator-catalog -n openshift-marketplace
```

2. If there is not a **spec.updateStrategy** section, or **spec.image** is not set to **icr.io/cpopen/ibm-operator-catalog:latest**, then update the YAML to have the following contents, and save it.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-operator-catalog
  publisher: IBM Content
  sourceType: grpc
  image: icr.io/cpopen/ibm-operator-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
```

3. Update foundational services

IBM Cloud Pak® foundational services, which is part of your IBM Cloud Pak for AIOps deployment, must be at version 3.23 or higher before you upgrade IBM Cloud Pak for AIOps.

Use the following steps to verify that your **ibm-common-service-operator** subscription is set to version 3.23 or higher, and to set it to a qualifying version if it is not.

1. Run the following command to find out what version of foundational services you have installed.

   ```
   oc get csv -A | grep ibm-common-service-operator
   ```

   If the version returned is v3.23 or higher, then you do not need to update foundational services and you must skip the rest of this section and proceed to step 4, Scale down the AIOpsEdge deployment.

2. Download the Common Services upgrade script, **upgrade_common_services.sh**, from github.com/IBM.

3. Run the following command from the directory that you downloaded the Common Services upgrade script to. This script must be run by a user with **cluster-admin** privilege.

   ```
   ./cp4waiops-samples/upgrade/upgrade_common_services.sh -a -c v3.23
   ```

   Important: You must only run this script if your version of foundational services is less than v3.23.

4. When **upgrade_common_services.sh** completes, run the following commands to verify that the **ibm-common-service-operator** channel is set to version 3.23 or higher in the subscription and in the ClusterServiceVersion (CSV) before you continue.

   1. Check the subscription.

      ```
      oc get subscription ibm-common-service-operator -n ibm-common-services -o
      jsonpath='{.spec.channel}'
      ```

      Example output:

      ```
      oc get subscription ibm-common-service-operator -n ibm-common-services -o
      jsonpath='{.spec.channel}'
      'v3.23'
      ```

   2. Check the CSV.

      ```
      oc get csv -A | grep ibm-common-service-operator
      ```

      Example output:

      ```
      oc get csv -A | grep ibm-common-service-operator
      ibm-common-services   ibm-common-service-operator.v3.23.7   IBM Cloud Pak
      foundational services   3.23.7   Succeeded
      ```

```
cp4aiops              ibm-common-service-operator.v3.23.7   IBM Cloud Pak
foundational services   3.23.7   Succeeded
```

5. The foundational services upgrade commences, and will take approximately 30 - 60 minutes.

   You can run the following command to check the status of **ZenService**. When the foundational services upgrade is complete, this command will have a *STATUS* of **Completed**. Do not proceed until the upgrade has completed.

   ```
   oc get zenservice -A -o custom-
   columns='KIND:.kind,NAME:.metadata.name,NAMESPACE:.metadata.namespace,VERSION:status.curr
   entVersion,STATUS:.status.zenStatus,PROGRESS:.status.Progress,MESSAGE:.status.ProgressMes
   sage'
   ```

   Example output from a successful foundational services upgrade:

   ```
   KIND          NAME                  NAMESPACE    VERSION    STATUS      PROGRESS    MESSAGE
   ZenService    iaf-zen-cpdservice    cp4waiops    4.8.0      Completed   100%        The
   Current Operation Is Completed
   ```

4. Scale down the AIOpsEdge deployment

1. Scale AIOpsEdge down to zero replicas.

   Run the following command:

   ```
   oc scale deployment aiopsedge-operator-controller-manager -n ${PROJECT_CP4AIOPS} --
   replicas=0
   ```

2. Wait for the AIOpsEdge operator pods to terminate.

   Do not proceed until the following command does not return any pods.

   ```
   oc get pod -n ${PROJECT_CP4AIOPS}  | egrep "aiopsedge-operator-controller-manager"
   ```

5. Update the operator subscription

Update the **spec.channel** value of the IBM Cloud Pak for AIOps subscription to the release that you want to upgrade to, v4.2.

```
oc patch subscription.operators.coreos.com ibm-aiops-orchestrator -n ${INSTALL_MODE_NAMESPACE}
--type=json -p='[{'op': 'replace', 'path': '/spec/channel', 'value': 'v4.2'}]'
```

6. Verify the deployment

6.1 Check the version

Run the following command and check that the *VERSION* that is returned is **4.2.0**.

```
oc get csv -l operators.coreos.com/ibm-aiops-orchestrator.${INSTALL_MODE_NAMESPACE} -n
${INSTALL_MODE_NAMESPACE}
```

Example output:

```
oc get csv -l operators.coreos.com/ibm-aiops-orchestrator.cp4aiops -n cp4aiops

NAME                             DISPLAY                    VERSION   REPLACES
PHASE
ibm-aiops-orchestrator.v4.2.0   IBM Cloud Pak for AIOps   4.2.0     ibm-aiops-orchestrator.v4.1.2
Succeeded
```

6.2 Check the deployment

Run the following command to check that the **PHASE** of your deployment is **Updating**.

```
oc get installations.orchestrator.aiops.ibm.com -n ${PROJECT_CP4AIOPS}
```

Example output:

```
NAME            PHASE      LICENSE    STORAGECLASS    STORAGECLASSLARGEBLOCK    AGE
ibm-cp-aiops    Updating   Accepted   rook-cephfs     rook-ceph-block           3m
```

It takes around 60-90 minutes for the upgrade to complete (subject to the speed with which images can be pulled). When installation is complete and successful, the **PHASE** of your installation changes to `Running`. If your installation phase does not change to `Running`, then use the following command to find out which components are not ready:

```
oc get installation.orchestrator.aiops.ibm.com -o yaml -n ${PROJECT_CP4AIOPS} | grep 'Not Ready'
```

Example output:

```
lifecycleservice: Not Ready
zenservice: Not Ready
```

To see details about why a component is `Not Ready` run the following command, where `<component>` is the component that is not ready, for example `zenservice`.

```
oc get <component> -o yaml -n ${PROJECT_CP4AIOPS}
```

(Optional) You can also download and run a status checker script to see information about the status of your deployment. For more information about how to download and run the script, see github.com/IBM.

If the installation fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade and Known Issues to help you identify any installation problems.

6.3 Verify that the LifecycleTrigger job is complete

Run the following command, and verify that the output contains `reason: Ready` and `status: "True"` before continuing to the next step.

```
oc get LifecycleTrigger aiops -o yaml
```

Example output:

```
status:
  conditions:
  - lastTransitionTime: "2024-03-21T03:04:53Z"
    message: Job '84086bada206a5b4b31cc297e3206835' in state 'RUNNING'
    observedGeneration: 2
    reason: Ready
    status: "True"
    type: LifecycleTriggerReady
```

7. Post upgrade actions

1. If you previously setup backup or restore on your deployment, then you must follow the instructions in Upgrading IBM Cloud Pak for AIOps backup and restore artifacts.
2. If the *EXPIRY_SECONDS* environment variable was set for configuring log anomaly alerts, the environment variable was not retained in the upgrade. After the upgrade is completed, set the environment variable again. For more information about setting the variable, see Configuring expiry time for log anomaly alerts.
3. If the `Access Control` page displays custom roles with deprecated permissions after upgrade, see Custom roles with deprecated permissions after upgrade.
4. If you have Netcool or Metrics connections configured, then you must follow the instructions in After upgrade, Netcool and Metrics connections have a red status in the Connections UI.

5. In a proactive ChatOps channel, if you click the Change request ticket URL, but details don't display in the ServiceNow instance, edit the information for the **`x_ibm_waiops.admin`** user. For more information, see IBM Change Risk Assessment tab in ServiceNow not displaying change risk assessment details.

Offline upgrade of IBM Cloud Pak for AIOps

Use these instructions to upgrade IBM Cloud Pak® for AIOps 4.1.0 or later to 4.2.0.

This procedure can be used on an offline deployment of IBM Cloud Pak for AIOps 4.1.0 or later, and can still be used if the deployment has had hotfixes applied. If you have an online deployment, follow the instructions in Upgrading IBM Cloud Pak for AIOps (online).

Before you begin

Notes:

- Ensure that you are on a version of Red Hat OpenShift that your current and target versions of IBM Cloud Pak for AIOps both support. If you already have a qualifying version of Red Hat OpenShift but you want to upgrade it, then complete the IBM Cloud Pak for AIOps upgrade first. For more information, see Guidance for upgrades that require an Red Hat OpenShift upgrade.
- Ensure that you are logged in to your Red Hat® OpenShift® Container Platform cluster with **`oc login`** for any steps that use the OpenShift command-line interface (CLI).
- Red Hat OpenShift Container Platform requires a user with **`cluster-admin`** privileges for the following operations:
  - Mirror images
  - Update the catalog
  - Update foundational services
  - Update the operator subscription

Warnings:

- Custom patches, labels, and manual adjustments to IBM Cloud Pak for AIOps resources are lost when IBM Cloud Pak for AIOps is upgraded, and must be manually reapplied after upgrade. For more information, see Manual adjustments are not persisted.
- If you previously increased the size of a PVC directly, then you must follow the correct procedure that is supplied in Adjusting PVCs to ensure that the size is updated by the operator. Failure to do so before upgrading IBM Cloud Pak for AIOps causes the operator to attempt to restore a lower default value for the PVC, and causes an error in your IBM Cloud Pak for AIOps deployment.

Restrictions:

- You cannot use these instructions to upgrade deployments of IBM Cloud Pak for AIOps 3.7.2 or earlier. For more information, see Upgrade paths.
- The upgrade cannot be removed or rolled back.

Upgrade procedure

Follow these steps to upgrade your offline IBM Cloud Pak for AIOps deployment.

1. Ensure cluster readiness
2. Download CASE files
3. Mirror images
4. Update the catalog
5. Update foundational services
6. Scale down AIOpsEdge and AIOps orchestrator
7. Update the operator subscription
8. Verify the deployment
9. Post upgrade actions

1. Ensure cluster readiness

Recommended: Take a backup before upgrading. For more information, see Backup and restore.

1. Ensure that your cluster still meets all of the prerequisites for an air-gapped deployment.

   Review the steps for your installation approach:

   - Bastion host: Prerequisites
   - Portable device: Prerequisites

   Note: IBM Cloud Pak for AIOps requires that Red Hat OpenShift Container Platform must be version 4.12.x.

2. Install the IBM Catalog Management Plug-in for IBM Cloud Pak®, if you do not have it installled yet.

   - Bastion host: Install the IBM Catalog Management Plug-in for IBM Cloud Pak®.
   - Portable device: Install the IBM Catalog Management Plug-in for IBM Cloud Pak®.

3. Download scripts.

   1. Download the prerequisite checker script and copy it to your air-gapped environment.

      For more information about the script, including how to download and run it, see github.com/IBM.

   2. Download the Common Services upgrade script, **`upgrade_common_services_airgap.sh`**, and copy it to your air-gapped environment.

      For more information about the script, including how to download and run it, see github.com/IBM.

   3. Download the IBM Cloud Pak for AIOps uninstall and cleanup script, and copy it to your air-gapped environment.

      For more information about the script, including how to download and run it, see github.com/IBM.

   4. (Optional) Download the status checker script, and copy it to your air-gapped environment.

      For more information about the script, including how to download and run it, see github.com/IBM. The status checker script can be used in step 9. Verify the deployment to give information about the status of your deployment. The use of this script is optional, as status can be found directly from the **`ibm-aiops-orchestrator`** custom resource.

4. Run the IBM Cloud Pak for AIOps prerequisite checker script.

   Run the prerequisite checker script to ensure that your Red Hat OpenShift Container Platform cluster is correctly set up for an IBM Cloud Pak for AIOps upgrade. When you run the prerequisite checker script, you must run the script in the same project (namespace) that IBM Cloud Pak for AIOps is installed in.

   Important: The prerequisite checker script might show inadequate resources in the *Resource Summary* because the script does not account for resources already being in use by the upgrading deployment. This can be ignored, as can the following message: *[FAIL] Small or Large Profile Install Resources*.

   Example output:

```
# ./prereq.sh
Using project "cp4waiops" on server "https://myserver.mycluster.mydomain:6443".

Starting IBM Cloud Pak for AIOps prerequisite checker v4.2...

[INFO] ================================Openshift Container Platform Version
Check================================
[INFO] Checking OCP Version. Compatible Versions of OCP are v4.12.x
[INFO] OCP version 4.12.18 is compaitble
[INFO] ================================Openshift Container Platform Version
Check================================

[INFO] ================================Entitlement Pull
Secret================================
[INFO] Checking whether the Entitlement secret or Global pull secret is configured
correctly.
```

```
[INFO] Checking if the job 'cp4waiops-entitlement-key-test-job' already exists.
[INFO] The job with name 'cp4waiops-entitlement-key-test-job' was not found, so moving
ahead and creating it.
[INFO] Creating the job 'cp4waiops-entitlement-key-test-job'
job.batch/cp4waiops-entitlement-key-test-job created
[INFO] Verifying if the job 'cp4waiops-entitlement-key-test-job' completed successfully..
[INFO] SUCCESS! Entitlement secret is configured correctly.
job.batch "cp4waiops-entitlement-key-test-job" deleted
[INFO] ================================Entitlement Pull
Secret================================

[INFO] ================================Storage Provider================================
[INFO] Checking storage providers
[INFO] No IBM Storage Fusion Found... Skipping configuration check.

[INFO] No Portworx StorageClusters found with "Running" or "Online" status. Skipping
configuration check for Portworx.
[INFO] Openshift Data Foundation found.
[INFO] No IBM Cloud Storage found... Skipping configuration check for IBM Cloud Storage
Check.

Checking Openshift Data Foundation Configuration...
Verifying if Red Hat Openshift Data Foundation pods are in "Running" or "Completed"
status
[INFO] Pods in openshift-storage project are "Running" or "Completed"
[INFO] ocs-storagecluster-ceph-rbd exists.
[INFO] ocs-storagecluster-cephfs exists.
[INFO] No warnings or failures found when checking for Storage Providers.
[INFO] ================================Storage Provider================================

[INFO] ================================Small or Large Profile Install
Resources================================
[INFO] Checking for cluster resources

[INFO] ================================Resource
Summary=========================================================
[INFO]                                        Nodes   |      vCPU      |     Memory(GB)
[INFO] Small profile(available/required)  [  14 / 3 ]   [  255 / 62 ]       [  479 / 140
]
[INFO] Large profile(available/required)  [  14 / 10 ]  [  255 / 162 ]      [  479 / 372
]
[INFO] ================================Resource
Summary=======================================
[INFO] Cluster currently has resources available to create a large profile of Cloud Pak
for AIOps
[INFO] ================================Small or Large Profile Install
Resources=================


[INFO] ================================Prerequisite Checker Tool
Summary================================
      [  PASS  ] Openshift Container Platform Version Check
      [  PASS  ] Entitlement Pull Secret
      [  PASS  ] Storage Provider
      [  PASS  ] Small or Large Profile Install Resources
[INFO] ================================Prerequisite Checker Tool
Summary================================
```

5. Set the environment variables that are needed for upgrade.

   If you still have **waiops_var.sh** from when you installed IBM Cloud Pak for AIOps, then run the following steps from the directory that the script is in:

   1. Change both occurrences of **PROJECT_CP4WAIOPS** in **waiops_var.sh** to be **PROJECT_CP4AIOPS**. Save your changes.

   2. Run the following command to source your script and set the environment variables.

      ```
      . ./waiops_var.sh
      ```

If you do not have **waiops_var.sh** from when you installed IBM Cloud Pak for AIOps, then run the following commands:

```
export PROJECT_CP4AIOPS=<project>
export INSTALL_MODE_NAMESPACE=<install_namespace>
```

Where

- **<project>** is the namespace (project) that your IBM Cloud Pak for AIOps subscription is deployed in.
- **<install_namespace>** is ${PROJECT_CP4AIOPS} if your deployment is namespace scoped, or **openshift-operators** if your deployment has a cluster-wide scope.

6. Delete any evicted **connector-orchestrator** pods.

    1. Run the following command to check if there are any evicted **connector-orchestrator** pods.

        ```
        oc get pods -n ${PROJECT_CP4AIOPS} | grep connector-orchestrator
        ```

    2. Cleanup any evicted connector-orchestrator pods.

        If the previous command returned any pods with a *STATUS* of **Evicted**, then run the following command to delete each of them.

        ```
        oc delete pod -n ${PROJECT_CP4AIOPS} <connector_orchestrator>
        ```

        Where **<connector_orchestrator>** is a pod returned in the previous step.

2. Download CASE files

On your OpenShift cluster, rerun step 2 of the air-gap installation procedure **Set environment variables and download CASE files** to download the latest CASE files.

Follow the steps for your installation approach:

- Bastion host: Download the CASE
- Portable device: Download the CASE

3. Mirror images

Rerun step 3 of the air-gap installation procedure to mirror the updated images to the offline registry.

Follow the steps for your installation approach:

- Bastion host: Mirror images
- Portable device: Mirror images

4. Update the catalog

Rerun step 5.1 of the air-gap installation procedure **Create the catalog source** to update your catalog source.

Follow the steps for your installation approach:

- Bastion host: Create the catalog source
- Portable device: Create the catalog source

5. Update foundational services

IBM Cloud Pak® foundational services, which is part of your IBM Cloud Pak for AIOps deployment, must be at version 3.23 or higher before you upgrade IBM Cloud Pak for AIOps.

Use the following steps to verify that your **ibm-common-service-operator** subscription is set to version 3.23 or higher, and to set it to a qualifying version if it is not.

Note: This section uses the scripts that you downloaded earlier in section 1, Ensure cluster readiness.

1. Run the following command to find out what version of foundational services you have installed.

   ```
   oc get csv -A | grep ibm-common-service-operator
   ```

   If the version returned is v3.23 or higher, then you do not need to update foundational services and you must skip the rest of this section and proceed to step 6, Scale down AIOpsEdge and AIOps orchestrator.

2. Run the following command from the directory that you downloaded the Common Services upgrade script to. This script must be run by a user with **cluster-admin** privilege.

   ```
   ./cp4waiops-samples/upgrade/upgrade_common_services_airgap.sh -a -c v3.23
   ```

   Important: You must only run this script if your version of foundational services is less than v3.23.

3. When **upgrade_common_services.sh** completes, run the following commands to verify that the **ibm-common-service-operator** channel is set to version 3.23 or higher in the subscription and in the ClusterServiceVersion (CSV) before you continue.

   1. Check the subscription.

      ```
      oc get subscription ibm-common-service-operator -n ibm-common-services -o
      jsonpath='{.spec.channel}'
      ```

      Example output:

      ```
      oc get subscription ibm-common-service-operator -n ibm-common-services -o
      jsonpath='{.spec.channel}'
      'v3.23'
      ```

   2. Check the CSV.

      ```
      oc get csv -A | grep ibm-common-service-operator
      ```

      Example output:

      ```
      oc get csv -A | grep ibm-common-service-operator
      ibm-common-services   ibm-common-service-operator.v3.23.7   IBM Cloud Pak
      foundational services   3.23.7   Succeeded
      cp4aiops              ibm-common-service-operator.v3.23.7   IBM Cloud Pak
      foundational services   3.23.7   Succeeded
      ```

4. The foundational services upgrade commences, and will take approximately 30 - 60 minutes.

   You can run the following command to check the status of **ZenService**. When the foundational services upgrade is complete, this command will have a *STATUS* of **Completed**. Do not proceed until the upgrade has completed.

   ```
   oc get zenservice -A -o custom-
   columns='KIND:.kind,NAME:.metadata.name,NAMESPACE:.metadata.namespace,VERSION:status.curr
   entVersion,STATUS:.status.zenStatus,PROGRESS:.status.Progress,MESSAGE:.status.ProgressMes
   sage'
   ```

   Example output from a successful foundational services upgrade:

   ```
   KIND          NAME                  NAMESPACE   VERSION   STATUS      PROGRESS   MESSAGE
   ZenService    iaf-zen-cpdservice    cp4waiops   4.8.0     Completed   100%       The
   Current Operation Is Completed
   ```

6. Scale down AIOpsEdge and AIOps orchestrator

1. Scale AIOpsEdge and AIOps orchestrator down to zero replicas.

Run the following commands:

```
oc scale deployment aiopsedge-operator-controller-manager -n ${PROJECT_CP4AIOPS} --
replicas=0
oc scale deployment ibm-aiops-orchestrator-controller-manager -n ${PROJECT_CP4AIOPS} --
replicas=0
```

2. Wait for the AIOpsEdge operator and AIOps orchestrator pods to terminate.

   Do not proceed until the following command does not return any pods.

   ```
   oc get pod -n ${PROJECT_CP4AIOPS}  | egrep "aiopsedge-operator-controller-manager|ibm-
   aiops-orchestrator-controller-manager"
   ```

7. Update the operator subscription

Run the following commands to update the IBM Cloud Pak for AIOps subscription and the **operandregistry** resources to use the new channel, v4.2.

```
oc project ${INSTALL_MODE_NAMESPACE}
oc patch operandregistry ibm-aiops --type=json -p='[
{'op': 'replace', 'path': '/spec/operators/0/channel', 'value': 'v4.2'},
{'op': 'replace', 'path': '/spec/operators/1/channel', 'value': 'v4.2'},
{'op': 'replace', 'path': '/spec/operators/2/channel', 'value': 'v4.2'},
{'op': 'replace', 'path': '/spec/operators/3/channel', 'value': 'v4.2'},
{'op': 'replace', 'path': '/spec/operators/4/channel', 'value': 'v4.2'},
{'op': 'replace', 'path': '/spec/operators/5/channel', 'value': 'v4.2'},
{'op': 'replace', 'path': '/spec/operators/7/channel', 'value': 'v4.2'},
{'op': 'replace', 'path': '/spec/operators/8/channel', 'value': 'v4.2'},
{'op': 'replace', 'path': '/spec/operators/9/channel', 'value': 'v4.2'},
{'op': 'replace', 'path': '/spec/operators/10/channel', 'value': 'v4.2'}]'
```

```
oc patch subscription.operators.coreos.com ibm-aiops-orchestrator -n ${INSTALL_MODE_NAMESPACE}
--type=json -p='[{'op': 'replace', 'path': '/spec/channel', 'value': 'v4.2'}]'
```

8. Verify the deployment

8.1 Check the version

Run the following command and check that the *VERSION* that is returned is **4.2.0**.

```
oc get csv -l operators.coreos.com/ibm-aiops-orchestrator.${INSTALL_MODE_NAMESPACE} -n
${INSTALL_MODE_NAMESPACE}
```

Example output:

```
oc get csv -l operators.coreos.com/ibm-aiops-orchestrator.cp4aiops -n cp4aiops
NAME                           DISPLAY                  VERSION  REPLACES
PHASE
ibm-aiops-orchestrator.v4.2.0  IBM Cloud Pak for AIOps  4.2.0    ibm-aiops-orchestrator.v4.1.2
Succeeded
```

8.2 Check the deployment

Run the following command to check that the **PHASE** of your deployment is **Updating**.

```
oc get installations.orchestrator.aiops.ibm.com -n ${PROJECT_CP4AIOPS}
```

Example output:

```
NAME           PHASE     LICENSE    STORAGECLASS    STORAGECLASSLARGEBLOCK   AGE
ibm-cp-aiops   Updating  Accepted   rook-cephfs     rook-ceph-block          3m
```

It takes around 60-90 minutes for the upgrade to complete (subject to the speed with which images can be pulled). When installation is complete and successful, the **PHASE** of your installation changes to **Running**. If your installation phase does not

change to **Running**, then use the following command to find out which components are not ready:

```
oc get installation.orchestrator.aiops.ibm.com -o yaml -n ${PROJECT_CP4AIOPS} | grep 'Not Ready'
```

Example output:

```
lifecycleservice: Not Ready
zenservice: Not Ready
```

To see details about why a component is **Not Ready** run the following command, where **<component>** is the component that is not ready, for example **zenservice**.

```
oc get <component> -o yaml -n ${PROJECT_CP4AIOPS}
```

(Optional) If you downloaded the status checker script earlier in step 1.3 Ensure cluster readiness, then you can also run this script to see information about the status of your deployment.

If the installation fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade and Known Issues to help you identify any installation problems.

8.3 Verify that the LifecycleTrigger job is complete

Run the following command, and verify that the output contains **reason: Ready** and **status: "True"** before continuing to the next step.

```
oc get LifecycleTrigger aiops -o yaml
```

Example output:

```
status:
  conditions:
  - lastTransitionTime: "2024-03-21T03:04:53Z"
    message: Job '84086bada206a5b4b31cc297e3206835' in state 'RUNNING'
    observedGeneration: 2
    reason: Ready
    status: "True"
    type: LifecycleTriggerReady
```

9. Post upgrade actions

1. Run the following command to delete the **ibm-vault-operator-catalog** catalog source, which is no longer required.

   ```
   oc delete catalogsource ibm-vault-operator-catalog -n openshift-marketplace
   ```

2. If you previously setup backup or restore on your deployment, then you must follow the instructions in Upgrading IBM Cloud Pak for AIOps backup and restore artifacts.

3. If the *EXPIRY_SECONDS* environment variable was set for configuring log anomaly alerts, the environment variable was not retained in the upgrade. After the upgrade is completed, set the environment variable again. For more information about setting the variable, see Configuring expiry time for log anomaly alerts.

4. If the **Access Control** page displays custom roles with deprecated permissions after upgrade, see Custom roles with deprecated permissions after upgrade.

5. If you have Netcool or Metrics connections configured, then you must follow the instructions in After upgrade, Netcool and Metrics connections have a red status in the Connections UI.

6. In a proactive ChatOps channel, if you click the Change request ticket URL, but details don't display in the ServiceNow instance, edit the information for the **x_ibm_waiops.admin** user. For more information, see IBM Change Risk Assessment tab in ServiceNow not displaying change risk assessment details.

Configuring automatic patch upgrades

Disable or enable the automatic application of patch upgrades on your online IBM Cloud Pak® for AIOps deployment.

If you have an online IBM Cloud Pak for AIOps deployment with catalog polling enabled and a new patch is available, then your deployment is automatically upgraded. If you do not want patch upgrades to occur automatically, then catalog polling must be disabled, and the image must be pinned to a specific version. For more information, see Controlling upgrade.

Disabling automatic patch upgrade

You can disable automatic patch upgrade at any time, using the Red Hat® OpenShift® Container Platform CLI or console.

Note: The **ibm-operator-catalog** also contains the catalogs for other IBM Cloud Paks®. If you have multiple IBM Cloud Paks installed on your cluster and you alter the catalog polling attribute, then automatic update is altered for all of your IBM Cloud Paks.

Disabling automatic patch upgrades with the OpenShift CLI

1. Run the following command to view and edit your **ibm-operator-catalog** CatalogSource instance.

   ```
   oc edit catalogsource ibm-operator-catalog -n openshift-marketplace
   ```

2. If there is a **spec.updateStrategy** section in the YAML like in the following example, then remove it and save your changes.

   ```
   updateStrategy:
     registryPoll:
       interval: 45m
   ```

3. Update the **ibm-operator-catalog** CatalogSource to fix it to use the current image digest, instead of **icr.io/cpopen/ibm-operator-catalog:latest**. This ensures that the **ibm-operator-catalog** CatalogSource pods do not pull the latest image if a node reload or other issue causes them to restart.

   Run the following commands:

   ```
   IMGDIGEST=`oc get pods -n openshift-marketplace -l=olm.catalogSource=ibm-operator-catalog
   --no-headers -o=jsonpath="{.items[0].status.containerStatuses[0].imageID}" -n openshift-
   marketplace`
   CATALOGIMG=`oc get catalogsource ibm-operator-catalog -n openshift-marketplace -o
   jsonpath='{.spec.image}'`
   if [ "${CATALOGIMG}" = "icr.io/cpopen/ibm-operator-catalog:latest" ]; then
     oc patch catalogsource ibm-operator-catalog -n openshift-marketplace --type=json -p "[{
   "op": "replace", "path": "/spec/image", "value": "\"$IMGDIGEST\"" }]"
   fi
   ```

Disabling automatic patch upgrades with the OpenShift console

1. Find the current image digest.

   1. Go to **Home** > **Projects**, and select **openshift-marketplace**.
   2. Click **Workloads** > **Pods**, and then search for **ibm-operator-catalog**.
   3. Click on the returned **ibm-operator-catalog-<...>** pod.
   4. Click **YAML** to switch to the YAML view.
   5. Search for **imageID** in the YAML, and copy down the value of **spec.containerStatuses.imageID**. The value will be in a format similar to the following:

      ```
      icr.io/cpopen/ibm-operator-catalog@sha256:<...>
      ```

2. Log in to your OpenShift cluster's console.

3. Go to **Administration** > **Cluster Settings**.
4. Click **Configuration**, and then scroll down and click **OperatorHub**.
5. Select **Sources**, and then scroll down and click **ibm-operator-catalog**.
6. Click **YAML** to switch to the YAML view.
7. If there is a **spec.updateStrategy** section in the YAML, as in the following example, then remove it and save your changes.

```
updateStrategy:
  registryPoll:
    interval: 45m
```

8. Set the value of **spec.image** to the value of the current image digest that you found in step 1, instead of to **icr.io/cpopen/ibm-operator-catalog:latest**, and save your changes. This ensures that the **ibm-operator-catalog** CatalogSource pods do not pull the latest image if a node reload or other issue causes them to restart.

Enabling automatic patch upgrade

You can enable automatic patch upgrade at any time, using the OpenShift CLI or console.

Use the following steps to configure your catalog to automatically poll for the latest images. This will cause patch upgrades to occur automatically, if a newer patch is available.

Note: **ibm-operator-catalog** also contains the catalogs for other IBM Cloud Paks. If you have multiple IBM Cloud Paks installed on your cluster and you enable the polling attribute, then automatic update is configured for all of them.

Enabling automatic patch upgrades with the OpenShift CLI

1. Run the following command to view and edit your **ibm-operator-catalog** CatalogSource instance.

```
oc edit catalogsource ibm-operator-catalog -n openshift-marketplace
```

2. If there is not a **spec.updateStrategy** section, or **spec.image** is not set to **icr.io/cpopen/ibm-operator-catalog:latest**, then update the YAML to have the following contents, and save it.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-operator-catalog
  publisher: IBM Content
  sourceType: grpc
  image: icr.io/cpopen/ibm-operator-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
```

Enabling automatic patch upgrades with the OpenShift console

1. Log in to your OpenShift cluster's console.
2. Go to **Administration** > **Cluster Settings**.
3. Click **Configuration**, and then scroll down and click **OperatorHub**.
4. Select **Sources**, and then scroll down and click **ibm-operator-catalog**.
5. Click **YAML** to switch to the YAML view.
6. If there is not a **spec.updateStrategy** section, or **spec.image** is not set to **icr.io/cpopen/ibm-operator-catalog:latest**, then update the YAML to have the following contents, and save it.

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: ibm-operator-catalog
  namespace: openshift-marketplace
spec:
  displayName: ibm-operator-catalog
  publisher: IBM Content
  sourceType: grpc
  image: icr.io/cpopen/ibm-operator-catalog:latest
  updateStrategy:
    registryPoll:
      interval: 45m
```

Upgrading Infrastructure Automation

Upgrade an existing online or offline deployment of Infrastructure Automation 4.1.0 or later to 4.2.0.

You cannot use these instructions to upgrade deployments of Infrastructure Automation 3.7.2 or earlier.

The upgrade procedure depends on whether your deployment is an online or offline (air-gapped) deployment.

- Online upgrade
- Offline upgrade

For more information about upgrading to Infrastructure Automation 4.2.0 from a podified installation of 3.5.x or older, you must follow these instructions to complete the required upgrade to PostgreSQL 13.

- Upgrading PostgreSQL for a podified Infrastructure management installation

Upgrade paths

Infrastructure Automation uses a versioning system of X.Y.Z, where X=version, Y=release, and Z=patch. For example, Infrastructure Automation 4.2.1 is version 4, release 2, patch 1. Infrastructure Automation upgrades can be release upgrades or patch upgrades.

You can upgrade by a maximum of one release at a time. If the release that you want to upgrade from is more than one release behind the release that you are upgrading to, then you must upgrade sequentially. For example, you cannot upgrade directly from Infrastructure Automation 3.6.0 to v4.2.0, you must use the following sequence:

1. Use the v3.7.2 documentation to upgrade from Infrastructure Automation 3.6.0 to v3.7.2.
2. Use the v4.1.2 documentation to upgrade from Infrastructure Automation 3.7.2 to v4.1.2.
3. Use the v4.2.0 documentation to upgrade from Infrastructure Automation 4.1.2 to v4.2.0.

Upgrading Infrastructure management appliances

The following topics describe the process of upgrading Infrastructure management appliances when a new release of Infrastructure Automation is available.

For more information about migrating from Red Hat CloudForms or IBM Cloud Pak for Multicloud Management appliances to Infrastructure Automation, see these instructions to complete a full appliance replacement upgrade.

- Migrating to an Infrastructure management appliance

For more information about upgrading Infrastructure management appliances, see:

- Upgrading Infrastructure management appliances

For more information about migrating from an appliance installation to a Kubernetes (podified) installation of Infrastructure management, see the following topic.

- Migrate from an appliance-based to a Kubernetes-based Infrastructure Management installation

Online upgrade of Infrastructure Automation (CLI method)

Use these instructions to upgrade Infrastructure Automation 4.1.0 or later to 4.2.0.

This procedure can be used on an online deployment of Infrastructure Automation 4.1.0 or later, and can still be used if the deployment has had hotfixes applied. If you have an offline deployment, follow the instructions in Offline upgrade of Infrastructure Automation.

Before you begin

Ensure that you are logged in to your OpenShift cluster with **`oc login`** for any steps that use the Red Hat® OpenShift® Container Platform command-line interface (CLI).

Restrictions

- You cannot use these instructions to upgrade deployments of Infrastructure Automation 3.7.2 or earlier. For more information, see Upgrade paths.
- The upgrade cannot be removed or rolled back.

Upgrade procedure

Follow these steps to upgrade your online Infrastructure Automation deployment.

1. Verify cluster readiness
2. Configure automatic catalog polling
3. Update the operator subscription
4. Verify the deployment
5. Post upgrade actions

1. Verify cluster readiness

1. Ensure that your cluster still meets all the prerequisites for deployment. For more information, see Prerequisites.
2. Run the following command to set the environment variable that you need for upgrade.

   ```
   export PROJECT_IA=<project>
   ```

   Where **`<project>`** is the namespace (project) that Infrastructure Automation is deployed in.

2. Configure automatic catalog polling

Verify that your catalog is set to automatically poll for updates. Your **`ibm-operator-catalog`** CatalogSource object can be configured to automatically poll for a newer version, and to retrieve it if one is available. This triggers an automatic update of your deployment. Polling for updates is enabled by configuring the following polling attribute:

```
spec.updateStrategy.registryPoll
```

You might have already elected to automatically accept product updates by adding the polling attribute to your ibm-operator-catalog YAML when you installed Infrastructure Automation, installed an Infrastructure Automation hotfix from IBM support, or when you installed another IBM Cloud Pak.

Use the following steps to check whether you already have a polling attribute set, and to configure it if you do not.

Note: **`ibm-operator-catalog`** also contains the catalogs for other IBM Cloud Paks. If you have multiple IBM Cloud Paks installed on your cluster, and you enable the polling attribute, then automatic update is configured for all of them.

1. Run the following command to view and edit your ibm-operator-catalog CatalogSource instance.

   ```
   oc edit catalogsource ibm-operator-catalog -n openshift-marketplace
   ```

2. If there is not a **`spec.updateStrategy`** section, then update the YAML to have the following contents, and save it.

```
updateStrategy:
  registryPoll:
    interval: 45m
```

3. Update the operator subscription

Use the following steps to update the spec.channel value of the Infrastructure Automation operator subscription to the release that you want to upgrade to.

1. Log in to your OpenShift cluster's console.
2. Select **Home > Search**.
3. From the Project list, select the project (namespace) that your Infrastructure Automation operator subscription is deployed in. This is your Infrastructure Automation project if your deployment is namespace scoped, or openshift-operators if your deployment has a cluster wide scope.
4. In the **Resources** list, select **SUB Subscription**. A list of subscriptions is displayed. Click the subscription called `ibm-infrastructure-automation-operator`. A new window with the subscription details for `ibm-infrastructure-automation-operator` is displayed.
5. Click the value in the **Update channel box**. A new window called **Change Subscription** update channel is displayed. Change the channel to v4.2 and click **Save**.
6. Repeat the above steps with subscription `ibm-management-cam-install` and `ibm-management-im-install`.

4. Verify the deployment

4.1 Check Infrastructure Automation upgrade status

Check the status of the Infrastructure Automation upgrade to make sure the PHASE is Installing and eventually it changes to Running.

```
oc get iaconfigs -n ${PROJECT_IA}
```

Example output:

```
NAME                PHASE         VERSION   STORAGECLASS   STORAGECLASSLARGEBLOCK   AGE
ibm-ia-installer    Running       4.2.0     rook-cephfs    rook-ceph-block          8m44s
```

Important: Do not proceed until the message that Infrastructure Automation upgrade is Running is displayed.

4.2 Check the version

Verify that the patch has been successfully installed.

1. Log in to your OpenShift cluster's console.
2. Click **Operators > Installed Operators.**
3. From the Project drop-down list, select the project (namespace) that Infrastructure Automation is deployed in.
4. Locate IBM Infrastructure Automation in the list and verify that the annotation underneath it shows 4.2.0.

4.3 Check the deployment

After a few minutes, run the following command from the CLI to verify that your Infrastructure Automation deployment is running again.

```
oc get iaconfigs.aiops.ibm.com -A
```

Example output:

```
NAME                PHASE         VERSION   STORAGECLASS   STORAGECLASSLARGEBLOCK   AGE
ibm-ia-installer    Running       4.2.0     rook-cephfs    rook-ceph-block          8m44s
```

5. Post upgrade actions

If you previously setup backup or restore on your deployment, then you must follow the instructions in Upgrading Infrastructure Automation backup and restore artifacts.

Offline upgrade of Infrastructure Automation

Use these instructions to upgrade Infrastructure Automation 4.1.0 or later to 4.2.0.

This procedure can be used on an offline deployment of Infrastructure Automation 4.1.0 or later, and can still be used if the deployment has had hotfixes applied. If you have an online deployment, follow the instructions in Online upgrade of Infrastructure Automation (CLI method).

Before you begin

Ensure that you are logged in to your OpenShift cluster with **`oc login`** for any steps that use the Red Hat® OpenShift® Container Platform command-line interface (CLI).

Restrictions

- You cannot use these instructions to upgrade deployments of Infrastructure Automation 3.7.2 or earlier. For more information, see Upgrade paths.
- The upgrade cannot be removed or rolled back.

Upgrade procedure

Follow these steps to upgrade your offline Infrastructure Automation deployment.

1. Verify prerequisites
2. Download CASE files
3. Mirror images
4. Create the catalog
5. Modify resources to use the new channel
6. Update the operator subscription
7. Verify the deployment
8. Post upgrade actions

1. Verify prerequisites

1. Ensure that your cluster still meets all the prerequisites for an air-gapped deployment.

   Review the steps for your installation approach:

   - Bastion host
   - Portable compute device
2. Run the following command to set the environment variable that you need for upgrade.

   ```
   export PROJECT_IA=<project>
   ```

   Where **`<project>`** is the namespace (project) that Infrastructure Automation is deployed in.

2. Download CASE files

On your OpenShift cluster, rerun step 2 of the air-gap installation procedure Set environment variables and download CASE files to download the latest CASE files.

Follow the steps for your installation approach:

- Bastion host
- Portable compute device
- Portable storage device

Note: If you have an offline cache from a previous installation, you have two options to ensure that the latest CASEs are saved when you are using a case save command. You must:

- Delete the existing CASE .tgz files from your offline cache.

- Run a case save by using a different offline cache directory, and then copy the contents of that new directory into your existing offline cache.

Once you have the latest files in your offline cache, you can continue the airgap process using the offline cache directory.

3. Mirror images

Rerun step 3 of the air-gap installation procedure to mirror the updated images to the offline registry. Follow the steps for your installation approach:

- Bastion host: 3 Mirror images to your final location
- Portable compute device: 3 Mirror images to portable registry
- Portable storage device: 3 Mirror images to portable registry

4. Create the catalog

Rerun step 5.2 of the air-gap installation procedure to update your catalog source.

Follow the steps for your installation approach:

- Bastion host
- Portable compute device
- Portable storage device

5. Modify resources to use the new channel

You can modify the resources to use the new channel with the following commands.

```
oc -n ${PROJECT_IA} patch operandregistry infrastructure-automation --type=json -p='[
   {'op': 'replace', 'path': '/spec/operators/0/channel', 'value': 'v4.2'},
   {'op': 'replace', 'path': '/spec/operators/1/channel', 'value': 'v4.2'}]'
```

6. Update the operator subscription

Update the **spec.channel** value of the IBM Cloud Pak® foundational services operator subscription to **v3.23**.

```
oc patch subscription ibm-common-service-operator -n ibm-common-services --type=json -
p='[{'op': 'replace', 'path': '/spec/channel', 'value': 'v3.23'}]'
```

Update the **spec.channel** value of the Infrastructure Automation operators subscription to the release that you want to upgrade to, such as v4.2.

```
oc patch subscription ibm-infrastructure-automation-operator -n ${PROJECT_IA} --type=json -
p='[{'op': 'replace', 'path': '/spec/channel', 'value': 'v4.2'}]'
```

```
oc patch subscription ibm-management-cam-install -n ${PROJECT_IA} --type=json -p='[{'op':
'replace', 'path': '/spec/channel', 'value': 'v4.2'}]'
```

```
oc patch subscription ibm-management-im-install -n ${PROJECT_IA} --type=json -p='[{'op':
'replace', 'path': '/spec/channel', 'value': 'v4.2'}]'
```

7. Verify the deployment

7.1 Check Infrastructure Automation upgrade status

Check the status of the Infrastructure Automation upgrade to make sure the PHASE is Installing and eventually it changes to Running.

```
oc get iaconfigs -n ${PROJECT_IA}
```

Example output:

```
NAME                PHASE         VERSION    STORAGECLASS    STORAGECLASSLARGEBLOCK    AGE
ibm-ia-installer    Running       4.2.0      rook-cephfs     rook-ceph-block           8m44s
```

Important: Do not proceed until the message that Infrastructure Automation upgrade is Running is displayed.

7.2 Check the version

Verify that the patch has been successfully installed.

1. Log in to your OpenShift cluster's console.
2. Click **Operators** > **Installed Operators**.
3. From the project drop-down list, select the project (namespace) that Infrastructure Automation is deployed in.
4. Locate IBM Infrastructure Automation in the list and verify that the annotation underneath it shows `4.2.0`.

7.3 Check the deployment

After a few minutes, run the following command from the CLI to verify that your Infrastructure Automation deployment is running again.

```
oc get iaconfigs.aiops.ibm.com -A
```

Example output:

```
NAME                PHASE         VERSION   STORAGECLASS    STORAGECLASSLARGEBLOCK   AGE
ibm-ia-installer    Running       4.2.0     rook-cephfs     rook-ceph-block          8m44s
```

Important: Do not proceed until the message that Infrastructure Automation upgrade is Running is displayed.

8. Post upgrade actions

If you previously setup backup or restore on your deployment, then you must follow the instructions in Upgrading Infrastructure Automation backup and restore artifacts.

Upgrading PostgreSQL for a podified Infrastructure management installation

If you are upgrading to Infrastructure Automation 4.2.0 from a podified installation of 3.6.x, you can follow the Online (CLI) or Offline instructions and skip these steps to upgrade to PostgreSQL 13.

If you are upgrading to Infrastructure Automation 4.2.0 from a podified installation of 3.5.x or older, you must follow these instructions to complete the required upgrade to PostgreSQL 13.

The process for upgrading PostgreSQL requires you to complete a database backup, upgrade your installation with the PostgreSQL image (PostgreSQL 13), and then restore your database by using the updated PostgreSQL.

Important: PostgreSQL 10 is not supported by IBM Cloud Pak for AIOps Infrastructure Automation. The following procedure uses PostgreSQL 10 as an example for upgrading PostgreSQL, specifically to PostgreSQL 13.

Prerequisites

- You must have a physical volume claim (PVC) attached to your existing PostgreSQL pod with sufficient space for upgrading. Run the following command in the PostgreSQL pod to ensure that the data directory is large enough to hold both the database backup file and the imported database:

  ```
  psql -d vmdb_production -c "SELECT pg_size_pretty(pg_database_size('vmdb_production'));"
  ```

  Consider allocating an extra 50% of space based on your current sizing since this command is only an estimate of size on disk. For instance, for 1 GB of space, increase your data directory to 1.5 GB. For 2 GB, increase to 3 GB.

- Increase the idle timeout value. The default timeout for **`oc logs/exec/rsh/`** commands is 15 minutes. If you do not increase this time, you might encounter issues with completing the migration steps as the debug pod can time out and force a restart of the process. For more information about changing this timeout setting, see How to change idle-timeout for oc logs/exec/rsh commands.

Procedure

1. Connect to your Red Hat OpenShift Container Platform cluster and run the following command to postpone the orchestrator operations so that you can back up the database without client connections:

   ```
   oc patch deployment orchestrator -p '{"spec":{"template":{"spec":{"containers":
   [{"name":"orchestrator","command":["sleep", "1d"]}]}}}}' -n <namespace>
   ```

   Where **<namespace>** is your project (namespace) where IBM Cloud Pak for AIOps is installed.

2. Remote shell into the database pod and run the database backup:

   ```
   oc rsh deploy/postgresql
   pg_dump -Fc -d vmdb_production > /path_to_large_data_volume
   exit
   oc cp --retries=-1 <postgresql pod>:/path_to_large_data_volume <your_backup_file>
   ```

   Store the backup on a volume outside of the PostgreSQL data directory, such as on a different physical volume.

3. Edit the Infrastructure management installation operand to update the PostgreSQL version (**postgresqlVersion:**) from **10** to **13**.

   You need to modify the **ibm-management-im-install** operand instance to change the **postgresqlVersion** key that has the value **10** to **13**.

   Then, save your changes. This change causes the existing PostgreSQL pod to stop and the new upgrade PostgreSQL (PostgreSQL 13) pod to start.

   The PostgreSQL pod restarts with PostgreSQL 13 and fails as the data directory is still configured for PostgreSQL 10. For more information about how the initial instance of the operand was created, see Deploying Infrastructure management as a containerized deployment (podified).

   Alternative method: You can use a command line to change the PostgreSQL version:

   1. Run the following command:

      ```
      oc get iminstalls.infra.management.ibm.com -n <namespace>
      ```

      Where **<namespace>** is the namespace where Infrastructure Automation is installed.

      Record the install name value. You need to use this value in the following step.

   2. Then, run the following command:

      ```
      oc edit iminstalls.infra.management.ibm.com/<install name> -n <namespace>
      ```

      Where **<install name>** is the name of the install that you recorded in the previous step.

   3. Find and modify the **postgresqlVersion** key that has the value **10** to `13 and save your changes.

4. Postpone Infrastructure management operations so that you can shut down the existing PostgreSQL pod and remove the existing PVC before a new PVC is created with the upgraded PostgreSQL. Run the following command:

   ```
   oc patch deployment ibm-infra-management-application -p '{"spec":{"template":{"spec":
   {"containers":[{"name":"ibm-infra-management-application","command":["sleep",
   "1d"]}]}}}}' -n <namespace>
   ```

   Where **<namespace>** is your project (namespace) where IBM Cloud Pak for AIOps is installed.

5. Store the database backup file outside of the physical volume that you are using for the PostgreSQL pod (outside of the data directory) to avoid any issues with deleting the existing PVC.

6. Delete the existing PVC:

   ```
   oc delete pvc/postgresql deploy/postgresql
   ```

7. Remove the configuration for postponing Infrastructure management operations:

   ```
   oc delete deploy/ibm-infra-management-application
   ```

   When this process completes, the PVC is re-created and the PostgreSQL pod is started.

8. Remote shell into the orchestrator container and clear the database:

```
oc rsh deploy/orchestrator
cd /var/www/miq/vmdb
source ./container_env
DISABLE_DATABASE_ENVIRONMENT_CHECK=1 rake db:drop db:create
exit
```

9. Copy the database backup into the PostgreSQL pod:

```
oc cp --retries=-1 <your_backup_file> <postgresql pod>:/var/lib/pgsql
```

10. Restore the database backup:

    1. Remote shell into the database pod:

       ```
       oc rsh deploy/postgresql
       ```

    2. Change directory to the directory where you copied the backup file.

       ```
       cd /var/lib/pgsql
       ```

    3. Run the restore in the background and capture the output in nohup.out. Ensure that the command begins with nohup. The last '&' is required to run the process in the background.

       ```
       nohup pg_restore -v -d vmdb_production <your_backup_file>&
       ```

    4. Review the nohup.out file and look for unignored errors. Use cat or less or more to search for errors.

       ```
       cat nohup.out
       ```

       When the process finishes, the error or warning count is reported if the process was successful. The following error is expected when you upgrade from PostgreSQL 10 to 13 and can be ignored:

       ```
       pg_restore: error: could not execute query: ERROR:  schema "public" already exists
       ```

       Review any other errors.

    5. Run ps to verify pg_restore is no longer running:

       ```
       ps -ef
       ```

    6. Remove the backup file from this pod if everything restored correctly. Keep the backup file stored elsewhere until you test and verify your upgraded application.

       ```
       rm -f <your_backup_file>
       exit
       ```

11. Remote shell into the debug orchestrator pod to migrate the database:

```
oc rsh deploy/orchestrator
cd /var/www/miq/vmdb
source ./container_env
rake db:migrate
exit
```

12. Delete the orchestrator deployment to remove the command override that was previously added:

```
oc delete deployment orchestrator -n <namespace>
```

Where **<namespace>** is your project (namespace) where IBM Cloud Pak for AIOps is installed.

The orchestrator is now redeployed and running with the upgraded PostgreSQL (PostgreSQL 13).

Migrating Red Hat CloudForms and IBM Cloud Pak for Multicloud Management appliances to Infrastructure

Automation

This topic describes the process of doing a full appliance replacement upgrade when a new release of Infrastructure Automation – Infrastructure Management is available.

The following migration paths are supported:

- IBM Cloud Pak for Multicloud Management 2.3.x - Infrastructure management to IBM Cloud Pak for AIOps 4.2.0 - Infrastructure Automation
- Red Hat CloudForms 4.7 through 5.0 to IBM Cloud Pak for AIOps 4.2.0 - Infrastructure Automation

Ordering migrations

You must complete all migrations of the databases before you configure replication and high availability. The reason is because you deploy new appliances for migrating to Infrastructure management.

Important: Do not run **evmserverd** if a database is in a partially migrated state.

Modifying the PostgreSQL configuration

This procedure modifies the strategy that is used for updating the PostgreSQL configuration. Previously, changes were made directly to the **postgresql.conf** file in the data directory. After you update the new configuration, changes will be made by editing a separate file in the **/etc/manageiq/postgresql.conf.d** directory. This modification allows future PostgreSQL configuration changes to be made more easily.

Note: The **01_miq_overrides.conf** file in the **/etc/manageiq/postgresql.conf.d** directory is reserved for application overrides and is re-created on application updates. **Important:** This file must not be edited. More files are read by the PostgreSQL server process according to the documentation here: PostgreSQL - Managing Configuration File Contents

1. Back up the **postgresql.conf** file so that you have any configuration changes:

   ```
   cp $APPLIANCE_PG_DATA/postgresql.conf <file_location>
   ```

2. Copy the updated **postgresql.conf** file to the **pg_data** directory:

   ```
   cp $APPLIANCE_SOURCE_DIRECTORY/TEMPLATE/var/opt/rh/rh-
   postgresql95/lib/pgsql/data/postgresql.conf $APPLIANCE_PG_DATA/postgresql.conf
   ```

3. Optional: If you updated the **postgresql.conf** file before this procedure, create a new **02_user_overrides.conf** file, and add any modifications that you have from the original **postgresql.conf** file. Example layout of the **02_user_overrides.conf** file:

   ```
   #--------------------------------------------
   # WRITE AHEAD LOG
   #--------------------------------------------

   wal_level = 'logical'
   wal_log_hints = on
   wal_buffers = 16MB
   min_wal_size = 1GB
   max_wal_size = 2GB
   checkpoint_completion_target = 0.9
   ```

   After the **02_user_overrides.conf** file is modified and saved, restart the postgresql service:

   ```
   systemctl restart $APPLIANCE_PG_SERVICE
   ```

All **postgresql.conf** customizations are now made to files in the **/etc/manageiq/postgresql.conf.d** directory due to the following **include_dir** line in the default configuration. The settings in these new files override ones in the **postgresql.conf** file.

```
include_dir = '/etc/manageiq/postgresql.conf.d`
```

Migrating to the Infrastructure Management appliance

Overview

This procedure describes the process of migrating to the Infrastructure management appliance. This procedure does not necessarily include migration of all possible customer modifications, so it is recommended that you fully test any modifications before you migrate your environment.

Important:

- Read through all of the steps in this procedure before you begin the migration process.
- Infrastructure management appliances require 12 GB memory. However, before you migrate your appliances, adjust resources in your environment to avoid performance issues.

You can classify the migration into three groups of appliances:

- VMDB appliance - An appliance with running workers that also hosts its own database that other appliances can also connect to.
- Non-VMDB appliance - An appliance with running workers that does not host a database. The appliance is connected to an external database.
- Dedicated database appliance - A Infrastructure management appliance or non-Infrastructure management virtual machine with no running workers on it: the appliance contains only a database for other appliances to connect to.

Migration workflow summary

In summary, the migration process follows this workflow:

1. Prepare appliances:

   Note: This step applies only for a multi-region Infrastructure management environment. Remove all the subscriptions from the global region before you migrate the global database since the subscriptions must be dropped and re-created.

2. Back up the database of your appliance.

3. Migrate appliances:

   1. Provision new appliances to replace each appliance.

   2. On the database and nondatabase appliances, copy the GUID file from the version of the same appliance to the same location.

   3. On the database and nondatabase appliances, use the appliance console menu to stop the evmserverd service, manually remove the pre-configured database and key, and fetch the v2_key from an appliance.

   4. Copy the appliance database backup file and restore the database on each VMDB and dedicated database appliance.

   5. Load the new version of the pglogical library on the VMDB and dedicated database appliances.

      Note: This step applies only for a multi-region Infrastructure management environment.

   6. Migrate the VMDB appliance databases and update the Automate Model.

   7. Restart **evmserverd** on the VMDB appliances.

4. Configure replication after the migration process is complete and appliances are running again.

Backing up current appliances

These steps do not affect the operations of your appliance infrastructure. However, they help ensure that you are able to roll back if required and replicate the network settings.

1. Find the size of the databases by using the following command. Then, ensure that your volume has enough space for the backup file.

   ```
   psql -d vmdb_production -c "SELECT pg_size_pretty(pg_database_size('vmdb_production'));"
   ```

2. If it's a multi-region environment, ensure you removed all subscriptions from the global database appliance.
3. Stop the **evmserverd** process.

   ```
   systemctl stop evmserverd
   ```

4. Back up the databases of your appliances in the root directory. Take a snapshot if possible.

   ```
   pg_dumpall -c --if-exists | gzip > vmdb.pg.gz
   ```

   Notes:

   - This command creates a compressed text-based backup. **pg_dumpall** is a utility for writing out ("dumping") all PostgreSQL databases of a cluster into one script file. For more information, see pg_dumpall.
   - When you run a **pg_dumpall**, the **pg_replication_slots** is not copied over causing the slots to become inactive. Also, the size of the global database appliance is reduced allowing for quicker backup and restore.
   - Even a compressed database backup can be large so consider storing it on a file system with adequate space such as /var/tmp. Review your file system available space by using the df command.
   - You must also provide backup for configuration files such as **/var/www/miq/vmdb/certs/v2_key**, **/var/www/miq/vmdb/config/database.yml**, and **/var/www/miq/vmdb/GUID**.
5. Back up the following files and take note of the appliance that each file comes from. These files are used to replace the old appliances in the database and for disaster recovery.

   - /var/www/miq/vmdb/GUID
6. During the upgrade, the iptables configuration file **/etc/sysconfig/iptables** is removed. If you changed the iptables configuration from the default, run the **iptables --list -n** command to see the current configuration. Run the following command to back up the iptables configuration:

   ```
   iptables-save > /etc/iptables.conf
   ```

   You can restore your iptables configuration file with the following command:

   ```
   iptables-restore < /etc/iptables.conf
   ```

   You can also add this command to **/etc/rc.local** to reload the rules at every restart.

Steps to follow on newly deployed Infrastructure management 4.2.0 Appliances

Complete the following steps on your appliances as specified to migrate to Infrastructure management.

Note: Some steps are run on certain appliances. Ensure you wait for each command to finish before you move to the next step.

1. Provision and start the new appliances.
2. On the VMDB and non-VMDB appliances, run the following commands as the root user:
   1. Copy the GUID file from the appliance.

      ```
      scp root@$IP_of_old_appliance:/var/www/miq/vmdb/GUID /var/www/miq/vmdb/
      ```

      Notes:

      - Complete these steps on VMDB appliances first. The non-VMDB appliances don't have a database to connect to if they're done first.
      - Complete this step only if you want to keep the same appliance/miq_servers records in the database and inherit that appliance's existing settings. If you plan on deploying new appliances with new IP addresses and hostname and not inherit the settings, you might prefer creating new records by not copying the GUID.
   2. Stop the evmserverd process.

      ```
      systemctl stop evmserverd
      ```

   3. Remove the previously generated encryption key.

      ```
      rm /var/www/miq/vmdb/certs/v2_key
      ```

   4. Copy the default database configuration file, ovewriting the previously generated one that used the deleted encryption key.

```
scp root@<IP_of_old_appliance>:/var/www/miq/vmdb/config/database.yml
/var/www/miq/vmdb/config/database.yml
```

5. Drop the previously generated database.

   ```
   dropdb vmdb_production
   ```

6. For VMDB appliances, configure the application, fetch the key from remote system, and Reset Configured Database.

7. For VMDB appliances, select the same region number and database password as the old installation. You can find the region number that is used for an installation that is listed as **Database/Region:** in the `appliance_console` summary information.

8. For non-VMDB appliances, configure the application by using the join region in external database option. Note as part of the database configuration step, fetch the `v2_key` from the appliances.

9. Allow the appliance to start.

3. On the new VMDB and new dedicated database appliances:

   1. Stop the evmserverd process.

      ```
      systemctl stop evmserverd
      ```

   2. Drop the default database.

      ```
      dropdb vmdb_production
      ```

   3. Copy the database dump from the appliance. Update the following location to reflect the source file location of the database dump and copy it to a location with enough available space.

      ```
      scp root@$IP_of_old_appliance:/var/tmp/vmdb.pg.gz /var/tmp
      ```

   4. Decompress and restore the database dump backup file.

      ```
      gunzip < vmdb.pg.gz | psql postgres
      ```

   5. Change to the vmdb directory.

      ```
      cd /var/www/miq/vmdb/
      ```

   6. Run the command appropriate to your environment to migrate everything in the database to work with the latest configuration. If you run into an error, contact IBM Support.

      ```
      rake db:migrate
      ```

4. Update the Automate Model to the latest version. This resets the ManageIQ and IBM domains (base domains) to a new and upgraded version. In a single (stand-alone database) or replicated environment, run this command on each VMDB appliance.

   ```
   rake evm:automate:reset
   ```

5. Start the evmserverd process. In a single (stand-alone database) or replicated environment, run this command on each VMDB appliance:

   ```
   systemctl start evmserverd
   ```

The migration process is now complete. Once appliances are running again, you can configure database replication in your environment. For more information, see Configuring Database Replication and Centralized Administration in the General Configuration guide.

Upgrading Infrastructure management appliances

The following steps describe the process of upgrading Infrastructure management appliances when a new release of IBM Cloud Pak for AIOps Infrastructure Automation is available.

The following upgrade paths are supported:

- IBM Cloud Pak for AIOps 3.3.x, 3.4.x, 3.5.x, or 3.6.x - Infrastructure Automation to IBM Cloud Pak for AIOps 3.7.1 - Infrastructure Automation

Before you begin

Important:

- If you are upgrading from Infrastructure Automation 3.6.x or newer, see the following guide.

  Upgrade an Infrastructure management appliance from 3.6.x

- If you are upgrading from Infrastructure Automation 3.5.x or older, see the instructions for each appliance type in the following guides.

- Read through all of the steps in this procedure before you begin the appliance upgrade process.

Infrastructure management has three types of appliances:

- VMDB appliance - An Infrastructure management appliance with running workers, which also hosts its own database that other appliances can connect to.

  Upgrade an Infrastructure management VMDB appliance

- Non-VMDB appliance - An Infrastructure management appliance with running workers that does not host a database. The appliance is connected to an external database.

  Upgrade an Infrastructure management non-VMDB appliance

- Dedicated database appliance - An Infrastructure management appliance or non-Infrastructure management virtual machine with no running workers on it. The appliance contains only a database for other appliances to connect to.

  Upgrade an Infrastructure management dedicated database appliance

Upgrade an Infrastructure management appliance from 3.6.x

The following steps describe the process of upgrading Infrastructure management appliances from 3.6.x.

Important: Do not run **`evmserverd`** if a database is in a partially migrated state.

1. Create a temporary upgrade directory.

   On the target appliance, create a temporary upgrade directory by running the following commands:

   ```
   mkdir /tmp/upgrade
   cd /tmp/upgrade
   ```

2. Stop the **`evmserverd`** process.

   Skip this step in dedicated database appliances:

   ```
   systemctl stop evmserverd
   ```

3. Download the IBM Cloud Pak for AIOps - Infrastructure Automation RPM updates image to the upgrade directory.

   For more information about downloading the image, see Passport Advantage part numbers. Download the RPM updates image and copy the image into the temporary upgrade directory.

4. Extract the .tgz on the target appliance.

   Run the following commands:

```
cd /tmp/upgrade
tar -zxvf infrastructure-management-upgrade-15.y*.tgz
```

5. Upgrade the RPM packages.

   Run the following commands:

   ```
   dnf -y upgrade --allowerasing --refresh *.rpm
   dnf -y autoremove
   ```

6. Migrate the database.

   Important: In a single (stand-alone database) or replicated environment, run the following command only on the first Non-VMDB appliance.

   Change to the vmdb directory.

   ```
   cd /var/www/miq/vmdb/
   ```

   Run the command appropriate to your environment to migrate everything in the database to work with the latest configuration. If you run into an error, contact IBM Support.

   ```
   rake db:migrate
   ```

7. Update the Automate Model to the latest version.

   This command resets the ManageIQ and IBM domains (base domains) to a new and upgraded version. Run the command appropriate to your environment to update the Automate Model:

   Important: In a single (stand-alone database) or replicated environment, run the following command only on the first Non-VMDB appliance.

   ```
   rake evm:automate:reset
   ```

8. Start the evmserverd process.

   Skip this step in dedicated database appliances:

   ```
   systemctl start evmserverd
   ```

9. Remove the temporary upgrade directory.

   Remove the upgrade directory by running the following command:

   ```
   rm -rf /tmp/upgrade
   ```

   The upgrade process is now complete. After appliances are running again, you can configure database replication in your environment. For more information, see Configuring Database Replication and Centralized Administration in the General Configuration guide.

Upgrade an Infrastructure management VMDB appliance

Important: Do not run `evmserverd` if a database is in a partially migrated state.

1. Create a temporary upgrade directory.

   On the target appliance, create a temporary upgrade directory by running the following commands:

   ```
   mkdir /tmp/upgrade
   cd /tmp/upgrade
   ```

2. Back up the current appliance database, v2_key, database.yml, and GUID file.

   Important: Some of these substeps in step 2 are only required if you are upgrading from Infrastructure management 3.5.x and older.

These steps don't affect the operations of your appliance infrastructure. However, they help ensure that you are able to roll back if required and replicate the network settings.

1. Find the size of the databases by using the following command. Then, ensure that your volume has enough space for the backup file.

   Important: This step is only required if you are upgrading from Infrastructure management 3.5.x and older.

   ```
   psql -d vmdb_production -c "SELECT
   pg_size_pretty(pg_database_size('vmdb_production'));"
   ```

2. If it's a multi-region environment, make sure that you removed all subscriptions from the global database appliance.

   Important: This step is only required if you are upgrading from Infrastructure management 3.5.x and older.

3. Stop the **evmserverd** process.

   ```
   systemctl stop evmserverd
   ```

4. Back up the databases of your appliances in the root directory. Take a snapshot if possible.

   Important: This step is only required if you are upgrading from Infrastructure management 3.5.x and older.

   ```
   pg_dump -Fc -d vmdb_production > <your_backup_file>.dump
   ```

   Notes:

   - If you changed your postgresql database configuration files, **pg_hba.conf**, and **postgresql.conf**, include these files when you back up the v2_key, database.yml, GUID files. These files need to be manually updated to include any changes you made previously.
   - The **pg_dump** command, referred to as **dump** in the **appliance_console** summary information, is another form of logical backup. This command can be used to dump a single database for the purpose of restoring the database at a later time. Just like **pg_dumpall**, the **pg_dump** command is not tied to the database version so that you can restore a dump from an older version into a new version. The **pg_dump** command does not capture system tables or other databases.
   - The **pg_dumpall** command creates a compressed text-based backup. **pg_dumpall** is a utility for writing out ("dumping") all PostgreSQL databases of a cluster into one script file. For more information, see pg_dumpall.
   - Running a **pg_dumpall** command the **pg_replication_slots** doesn't get copied over causing the slots to become inactive. This action helps to reduce the size of the global database appliance and allows for quicker backup and restore.
   - Even a compressed database backup can be large so consider storing it on a file system with adequate space such as /var/tmp. Review your available file system space by using the **df** command.
   - You must also provide backup for configuration files such as **/var/www/miq/vmdb/certs/v2_key**, **/var/www/miq/vmdb/config/database.yml**, and **/var/www/miq/vmdb/GUID**.

5. Back up the following files and make a note of the appliance that each file comes from. This step is to replace the old appliances in the database and for disaster recovery.

   Important: This step is only required if you are upgrading from Infrastructure management 3.5.x and older.

   Note: The REGION file might not exist.

   - **/var/www/miq/vmdb/GUID**
   - **/var/www/miq/vmdb/REGION**

3. Clear the data directory.

   Important: This step is only required if you are upgrading from Infrastructure management 3.5.x and older.

   After the database and postgresql are safely backed up, clear the data directory by running the following commands:

   ```
   systemctl stop postgresql
   rm -rf $APPLIANCE_PG_DATA
   ```

4. Download the IBM Cloud Pak for AIOps - Infrastructure Automation RPM updates image to the upgrade directory.

For more information about downloading the image, see Passport Advantage part numbers. Download the RPM updates image and copy the image into the temporary upgrade directory.

5. Extract the .tgz on the target appliance.

   Run the following commands:

   ```
   cd /tmp/upgrade
   tar -zxvf infrastructure-management-upgrade-15.y*.tgz
   ```

6. Enable new stream 8 modules.

   Important: This step is only required if you are upgrading from Infrastructure management 3.5.x and older.

   Enable the stream 8 modules by running the following commands:

   ```
   dnf -y module reset postgresql
   dnf -y module enable postgresql:13
   dnf -y module reset ruby
   dnf -y module enable ruby:3.0
   ```

7. Upgrade the RPM packages.

   Run the following commands:

   ```
   dnf -y upgrade --allowerasing --refresh *.rpm
   dnf -y autoremove
   ```

8. Initialize the data directory.

   Important: This step is only required if you are upgrading from Infrastructure management 3.5.x and older.

   Run the following command:

   ```
   appliance_console_cli --internal --standalone --password smartvm
   ```

9. Verify that the PostgreSQL version is 13 by running the command:

   Important: This step is only required if you are upgrading from Infrastructure management 3.5.x and older.

   ```
   cat $APPLIANCE_PG_DATA/PG_VERSION
   ```

   { :codeblock}

   This command checks the PostgreSQL version and returns: 13

10. Restore the database with PostgreSQL 13 by using the command:

    Important: This step is only required if you are upgrading from Infrastructure management 3.5.x and older.

    ```
    pg_restore -v -d vmdb_production <your_backup_file>.dump
    ```

    { :codeblock}

11. Migrate the database and reset the ManageIQ and IBM automate domains.

    Change to the vmdb directory.

    ```
    cd /var/www/miq/vmdb/
    ```

    Run the command appropriate to your environment to migrate everything in the database to work with the latest configuration. If you run into an error, contact IBM Support.

    ```
    rake db:migrate
    ```

    Update the Automate Model to the latest version. This command resets the ManageIQ and IBM domains (base domains) to a new and upgraded version. Run the command appropriate to your environment to update the Automate Model:

    In a single (stand-alone database) or replicated environment, run the following command on each VMDB appliance.

    ```
    rake evm:automate:reset
    ```

12. Start the evmserverd process.

    In a single (stand-alone database) or replicated environment, run this command on each VMDB appliance:

    ```
    systemctl start evmserverd
    ```

13. Remove the temporary upgrade directory.

    Remove the upgrade directory by running the following command:

    ```
    rm -rf /tmp/upgrade
    ```

    The upgrade process is now complete. After appliances are running again, you can configure database replication in your environment. For more information, see Configuring Database Replication and Centralized Administration in the General Configuration guide.

Upgrade an Infrastructure management Non-VMDB appliance

Important: Do not run **evmserverd** if a database is in a partially migrated state.

1. Create a temporary upgrade directory.

   On the target appliance, create a temporary upgrade directory by running the following commands:

   ```
   mkdir /tmp/upgrade
   cd /tmp/upgrade
   ```

2. Stop the evmserverd process.

   Run the following command:

   ```
   systemctl stop evmserverd
   ```

3. Back up important files.

   Important: This step is only required if you are upgrading from Infrastructure management 3.5.x and older.

   Back up the following files and make a note of the appliance that each file comes from. This step is to replace the old appliances in the database and for disaster recovery.

   Note: The REGION file might not exist.

   - /var/www/miq/vmdb/GUID
   - /var/www/miq/vmdb/REGION

4. Download the IBM Cloud Pak for AIOps - Infrastructure Automation RPM updates image to the upgrade directory.

   For information on downloading the image, see Passport Advantage part numbers. Download the RPM updates image and copy the image into the temporary upgrade directory.

5. Extract the .tgz on the target appliance.

   Run the following commands:

   ```
   cd /tmp/upgrade
   tar -zxvf infrastructure-management-upgrade-15.y*.tgz
   ```

6. Enable new stream 8 modules.

   Important: This step is only required if you are upgrading from Infrastructure management 3.5.x and older.

   Enable the stream 8 modules by running the following commands:

   ```
   dnf -y module reset postgresql
   dnf -y module enable postgresql:13
   dnf -y module reset ruby
   dnf -y module enable ruby:3.0
   ```

7. Upgrade the RPM packages.

   Run the following commands:

   ```
   dnf -y upgrade --allowerasing --refresh *.rpm
   dnf -y autoremove
   ```

8. Migrate the database.

   Important: In a single (stand-alone database) or replicated environment, run the following command only on the first Non-VMDB appliance.

   Change to the vmdb directory.

   ```
   cd /var/www/miq/vmdb/
   ```

   Run the command appropriate to your environment to migrate everything in the database to work with the latest configuration. If you receive an error, contact IBM Support.

   ```
   rake db:migrate
   ```

9. Update the Automate Model to the latest version.

   This command resets the ManageIQ and IBM domains (base domains) to a new and upgraded version. Run the command appropriate to your environment to update the Automate Model:

   Important: In a single (stand-alone database) or replicated environment, run the following command only on the first Non-VMDB appliance.

   ```
   rake evm:automate:reset
   ```

10. Start the evmserverd process.

    In a single (stand-alone database) or replicated environment, run this command on each Non-VMDB appliance:

    ```
    systemctl start evmserverd
    ```

11. Remove the temporary upgrade directory.

    Remove the upgrade directory by running the following command:

    ```
    rm -rf /tmp/upgrade
    ```

    The upgrade process is now complete. After appliances are running again, you can configure database replication in your environment. For more information, see Configuring Database Replication and Centralized Administration in the General Configuration guide.

Upgrade an Infrastructure management Dedicated database appliance

1. Create a temporary upgrade directory.

   On the target appliance, create a temporary upgrade directory by running the following commands:

   ```
   mkdir /tmp/upgrade
   cd /tmp/upgrade
   ```

2. Back up the current appliance database.

   Important: All of these substeps in step 2 are only required if you are upgrading from Infrastructure management 3.5.x and older.

   These steps don't affect the operations of your appliance infrastructure. However, they help ensure that you are able to roll back if required and replicate the network settings.

1. Find the size of the databases by using the following command. Then, ensure that your volume has enough space for the backup file.

   Important: This step is only required if you are upgrading from Infrastructure management 3.5.x and older.

   ```
   psql -d vmdb_production -c "SELECT
   pg_size_pretty(pg_database_size('vmdb_production'));"
   ```

2. If it's a multi-region environment, make sure that you removed all subscriptions from the global database appliance.

   Important: This step is only required if you are upgrading from Infrastructure management 3.5.x and older.

3. Back up the databases of your appliances in the root directory. Take a snapshot if possible.

   Important: This step is only required if you are upgrading from Infrastructure management 3.5.x and older.

   ```
   pg_dumpall -c --if-exists | gzip > vmdb.pg.gz
   ```

   Notes:

   - If you changed your postgresql database configuration files, such as the pg_hba.conf and postgresql.conf then update these files manually to include any changes you made previously.
   - The `pg_dumpall` command creates a compressed text-based backup. `pg_dumpall` is a utility for writing out ("dumping") all PostgreSQL databases of a cluster into one script file. For more information, see pg_dumpall.
   - Running a `pg_dumpall` command the `pg_replication_slots` doesn't get copied over causing the slots to become inactive. This action helps to reduce the size of the global database appliance and allows for quicker backup and restore.
   - Even a compressed database backup can be large so consider storing it on a file system with adequate space such as /var/tmp. Review your file system available space by using the `df` command.

3. Clear the data directory.

   Important: This step is only required if you are upgrading from Infrastructure management 3.5.x and older.

   After the database and postgresql are safely backed up, clear the data directory by running the following commands:

   ```
   systemctl stop postgresql
   rm -rf $APPLIANCE_PG_DATA
   ```

4. Download the IBM Cloud Pak for AIOps - Infrastructure Automation RPM updates image to the upgrade directory.

   For more information, see Passport Advantage part numbers. Download the IBM Cloud Pak for AIOps 4.2.0 - Infrastructure Automation RPM updates image.

5. Extract the .tgz on the target appliance.

   Run the following commands:

   ```
   cd /tmp/upgrade
   tar -zxvf infrastructure-management-upgrade-15.y*.tgz
   ```

6. Enable new stream 8 modules.

   Important: This step is only required if you are upgrading from Infrastructure management 3.5.x and older.

   Enable the stream 8 modules by running the following commands:

   ```
   dnf -y module reset postgresql
   dnf -y module enable postgresql:13
   dnf -y module reset ruby
   dnf -y module enable ruby:3.0
   ```

7. Upgrade the RPM packages.

   Run the following commands:

   ```
   dnf -y upgrade --allowerasing --refresh *.rpm
   dnf -y autoremove
   ```

8. Verify that the PostgreSQL version is 13.

   Important: This step is required only if you are upgrading from Infrastructure management 3.5.x and older.

   Run the following command:

   ```
   cat $APPLIANCE_PG_DATA/PG_VERSION
   ```

   This command checks the PostgreSQL version and returns: 13

9. Complete the steps in Upgrading Non-VMDB appliances.

   Complete the steps in Upgrade an Infrastructure management non-VMDB appliance

10. Remove the temporary upgrade directory.

    Remove the upgrade directory by running the following command:

    ```
    rm -rf /tmp/upgrade
    ```

    The upgrade process is now complete. After appliances are running again, you can configure database replication in your environment. For more information, see Configuring Database Replication and Centralized Administration in the General Configuration guide.

Migrating from an Infrastructure management appliance to a Kubernetes (podified) installation

Follow these steps to migrate from an appliance installation to a Kubernetes (podified) installation of Infrastructure management.

Before you begin

You must have an Infrastructure management appliance that is installed and a containerized (podified) installation of Infrastructure management.

Collect data from the Infrastructure management appliance

1. Take a backup of the database from the appliance.

   ```
   pg_dump -Fc -d <database_name> > /root/pg_dump
   ```

   Where **<database_name>** is **vmdb_production** by default. Find your database name by running **appliance_console** on your appliance.

2. Export the encryption key and Base64 encode it for the Kubernetes Secret.

   ```
   vmdb && rails r "puts Base64.encode64(ManageIQ::Password.key.to_s)"
   ```

3. Get the region number.

   ```
   vmdb && rails r "puts MiqRegion.my_region.region"
   ```

4. Get the GUID of the server that you want to run as.

   ```
   vmdb && cat GUID
   ```

Restore the backup into the Kubernetes environment

Important:

Two options are available. You can restore the backup of the appliance into a new installation or you can restore the backup into an existing installation.

- Restore the appliance backup into a new Kubernetes based installation
- Restore the appliance backup into an existing Kubernetes based installation

Restore the backup into a new Kubernetes based installation

For a new installation, you need to make sure you configured OIDC for integration with IBM Cloud Paks®. You must do so before you create the yaml file and continue with this procedure. Follow the steps in OIDC Configuration Steps for Infrastructure Management.

1. Create a YAML file to define the Custom Resource (CR). Minimally you need the following.

```
apiVersion: infra.management.ibm.com/v1alpha1
kind: IMInstall
metadata:
  name: im-iminstall
spec:
  applicationDomain: inframgmtinstall.apps.mycluster.com
  databaseRegion: <region number from the appliance>
  serverGuid: <GUID value from appliance>
```

   Example CR

```
apiVersion: infra.management.ibm.com/v1alpha1
kind: IMInstall
metadata:
  labels:
    app.kubernetes.io/instance: ibm-infra-management-install-operator
    app.kubernetes.io/managed-by: ibm-infra-management-install-operator
    app.kubernetes.io/name: ibm-infra-management-install-operator
  name: im-iminstall
  namespace: c4waiops
spec:
  applicationDomain: inframgmtinstall.apps.mycluster.com
  databaseRegion: '99'
  serverGuid: <GUID value from appliance above>
  imagePullSecret: my-pull-secret
  enableSSO: true
  initialAdminGroupName: inframgmtusers
  license:
    accept: true
```

2. Create the CR in your namespace. The operator creates several more resources and starts deploying the app.

```
oc create -f <CR file name>
```

3. Edit the secret and insert the encryption key from the appliance. Replace the "encryption-key" value with the value exported from the appliance.

```
oc edit secret app-secrets -n c4waiops
```

4. Temporarily hijack the orchestrator by adding the following to the deployment:

```
oc patch deployment orchestrator -p '{"spec":{"template":{"spec":{"containers":
[{"name":"orchestrator","command":["sleep", "1d"]}]}}}}' -n cp4aiops
```

   Note:

   The default timeout for **oc debug** is 1 minute of inactivity. Increase the idle timeout values for the client and server. Otherwise, you might have difficulty in completing the remaining steps without the debug pod timing out and forcing a restart of the entire process. For more information, see How to change idle-timeout for oc commands like oc rsh, oc logs, and oc port-forward in OCP4.

5. Shell into the orchestrator container:

```
oc rsh deploy/orchestrator
```

   Clear postgres database:

```
cd /var/www/miq/vmdb
source ./container_env
DISABLE_DATABASE_ENVIRONMENT_CHECK=1 rake db:drop db:create
```

6. Copy your appliance database file into the postgresql pod. You must run the command from your cluster vm, not from within the pod.

```
oc cp <your_backup_file> <postgresql pod>:/var/lib/pgsql
```

7. Run the command **oc rsh** into the database pod and restore the database backup.

```
cd /var/lib/pgsql
# --- you copied the backup here in step 6. ---
pg_restore -v -d vmdb_production <your_backup_file>
rm -f <your_backup_file>
exit
```

8. Back in the debug pod from step 5:

```
rake db:migrate
exit
```

9. Delete the orchestrator deployment to remove the command override that was previously added:

```
oc delete deployment orchestrator -n cp4aiops
```

The orchestrator starts deploying the rest of the pods that are required to run Infrastructure management.

Restore the backup into an existing Kubernetes based installation

1. For an existing installation, find the existing CR by running the command,

```
oc get IMInstall -n cp4aiops
```

2. Edit the **IMInstall** resource from the CLI with the details in the previous step, for example, **oc edit IMInstall im-install** and add the following lines under **spec:**

```
  databaseRegion: <region number from the appliance>
  serverGuid: <GUID value from appliance>
```

   OR

   Log in to Red Hat OpenShift Container Platform for your cluster. Click **Home > Search**. In the **Resources** list menu, find **IMInstall**. Click the **IMInstall** resource; then, click the **YAML** tab. Add the following lines under **spec:**

```
  databaseRegion: <region number from the appliance>
  serverGuid: <GUID value from appliance>
```

   Click **Save**

3. Edit the secret and insert the encryption key from the appliance. Replace the "encryption-key" value with the value exported from the appliance.

```
oc edit secret app-secrets -n cp4aiops
```

4. Temporarily hijack the orchestrator by adding the following to the deployment:

```
oc patch deployment orchestrator -p '{"spec":{"template":{"spec":{"containers":
[{"name":"orchestrator","command":["sleep", "1d"]}]}}}}' -n cp4aiops
```

   Note: The default timeout for **oc debug** is 1 minute of inactivity. Increase the idle timeout values for the client and server. Otherwise, you might have difficulty in completing the remaining steps without the debug pod timing out and forcing a restart of the entire process. For more information, see How to change idle-timeout for oc commands like oc rsh, oc logs, and oc port-forward in OCP4.

5. Shell into the orchestrator container:

```
oc rsh deploy/orchestrator
```

   Clear postgres database:

```
cd /var/www/miq/vmdb
source ./container_env
DISABLE_DATABASE_ENVIRONMENT_CHECK=1 rake db:drop db:create
```

6. Copy your appliance database file into the postgresql pod. You must run the command from your cluster vm, not from within the pod.

```
oc cp <your_backup_file> <postgresql pod>:/var/lib/pgsql
```

7. Run the command **`oc rsh`** into the database pod and restore the database backup.

```
cd /var/lib/pgsql
# --- you copied the backup here in step 6. ---
pg_restore -v -d vmdb_production <your_backup_file>
rm -f <your_backup_file>
exit
```

8. Back in the debug pod from step 5:

```
rake db:migrate
exit
```

9. Delete the orchestrator deployment to remove the command override that was previously added:

```
oc delete deployment orchestrator -n cp4aiops
```

Important:

If your Infrastructure management instance on your cluster was previously configured for OIDC/SSO, you need to complete these additional steps.

1. Log in to IBM Cloud Paks®, and from the menu, click **Automate infrastructure > Infrastructure Management**. SSO to Infrastructure management is expected to fail.
2. On the Infrastructure management login screen, log in with the default credentials, user **`admin`**, and password **`smartvm`**.
3. In Infrastructure management, click **Settings > Application Settings**. Expand **Access Control** and click **Groups**.
4. Click **Configuration > Add a new Group**, and create a group with the following fields:
   - *Description:* **`<LDAP_group_name>`**
   - *Role:* **`EvmRole-super-administrator`**
   - *Project/Tenant:* **`<Your_tenant>`**

   Where **`<LDAP_group_name>`** is the value for **`initialAdminGroupName:`** found in the installation CR.
5. When finished, click **Add**.
6. Expand **Settings** and click **Zones > Server: EVM**. Click the **Authentication** tab, and ensure that the following settings are displayed:
   - *Mode*: External (**`httpd`**)
   - *Enable Single Sign-On*: checked
   - *Provider Type*: Enable OpenID-Connect
   - *Get User Groups from External Authentication (httpd)*: checked
7. Log out of Infrastructure management and IBM Cloud Paks®. Log in to IBM Cloud Paks® by using **Enterprise LDAP** authentication. From the menu, click **Automate infrastructure > Infrastructure Management**. SSO to Infrastructure management can now succeed.

Customizing the installation

Configuring the application domain name

It is necessary to set the **`applicationDomain`** in the CR if you are running in a production cluster. For a Code Ready Containers cluster, it might be something like **`miqproject.apps-crc.testing`**

External postgresql

Running with an external Postgres server is an option, if you want the default internal Postgres you can skip this step. Additionally, if you want to secure the connection, you need to include the optional parameters **sslmode=verify-full** and **rootcertificate** when you create the secret. To complete this process, manually create the secret and substitute your values before you create the CR.

```
oc create secret generic postgresql-secrets \
  --from-literal=dbname=vmdb_production \
  --from-literal=hostname=<your hostname> \
  --from-literal=port=5432 \
  --from-literal=password=<your password> \
  --from-literal=username=<your username> \
  --from-literal=sslmode=verify-full \ # optional
  --from-file=rootcertificate=path/to/optional/certificate.pem # optional
```

Note: If wanted, the secret name is also customizable by setting **databaseSecret** in the CR.

Using TLS to encrypt connections between pods inside the cluster:

Create a secret that contains all of the certificates

Certificates must be signed by a CA and that CA certificate must be in uploaded as **root_crt** so that it can be used to verify connection validity. If the secret is named **internal-certificates-secret**, no changes are needed in the CR, if you choose a different name that must be set in the **internalCertificatesSecret** field of the CR.

```
    oc create secret generic internal-certificates-secret \
      --from-file=root_crt=./certs/root.crt \
      --from-file=httpd_crt=./certs/httpd.crt \
      --from-file=httpd_key=./certs/httpd.key \
      --from-file=kafka_crt=./certs/kafka.crt \
      --from-file=kafka_key=./certs/kafka.key \
      --from-file=memcached_crt=./certs/memcached.crt \
      --from-file=memcached_key=./certs/memcached.key \
      --from-file=postgresql_crt=./certs/postgresql.crt \
      --from-file=postgresql_key=./certs/postgresql.key \
      --from-file=api_crt=./certs/api.crt \
      --from-file=api_key=./certs/api.key \
      --from-file=remote-console_crt=./certs/remote-console.crt \
      --from-file=remote-console_key=./certs/remote-console.key \
      --from-file=ui_crt=./certs/ui.crt \
      --from-file=ui_key=./certs/ui.key
```

Generating certificates:

A certificate authority (CA) and certificates are needed for each service. These certificates need to be valid for the internal Kubernetes service name, such as **httpd** or **postgres**. The services (UI, api) that back the route require the certificate to include a SAN with the application domain name. For example, the certificate for the UI needs to be valid for the hostname **ui** and **your_application.apps.example.com**.

Configuring external messaging

It is possible to run with an external Kafka messaging server. To do so, create the required secret with the correct parameters by using the following example and provide that secret name as **kafkaSecret** in the CR.

```
kubectl create secret generic kafka-secrets --from-literal=hostname=<your fqdn> --from-
literal=username=<your username> --from-literal=password=<your password>
```

If you decide to use a secret name other than **kafka-secrets**, you need to specify that in the CR.

```
kafkaSecret: <name of your secret>
```

Creating a custom TLS secret

You can use a custom TLS certificate. To use a custom certificate create the required secret with the correct parameters by using the folowing example and provide that secret name as **tlsSecret** in the CR.

```
oc create secret tls tls-secret --cert=tls.crt --key=tls.key`
```

If you decide to use a secret name other than **tls-secret**, you need to specify that in the CR.

```
tlsSecret: <name of your tls secret>
```

Configuring an image pull secret

If authentication is required to pull the images, a secret that contains the credentials needs to be created. The following script is an example of creating the secret.

```
kubectl create secret docker-registry image-pull-secret --docker-username=<your registry
username> --docker-password=<your registry password> --docker-server=<your registry server>
```

If you decide to use a secret name other than **image-pull-secret**, you need to specify that in the CR.

```
imagePullSecret: <name of your pull secret>
```

Configuring OpenID-Connect Authentication

To run with OpenID-Connect Authentication, customize the following example as required to fit your environment. For the following example, we tested with Keycloak version 11.0

Create a secret that contains the OpenID-Connect's **Client ID** and **Client Secret**. The values for **CLIENT_ID** and **CLIENT_SECRET** come from your authentication provider's client definition.

```
oc create secret generic <name of your kubernetes secret> --from-literal=CLIENT_ID=<your auth
provider client ID> --from-literal=CLIENT_SECRET=<your auth provider client secret>
```

Modify the CR with the following values:

```
httpdAuthenticationType: openid-connect
oidcProviderURL: https://<your keycloak FQDN>/auth/realms/<your realm>/.well-known/openid-
configuration
oidcClientSecret: <name of your kubernetes secret>
```

Configuring OpenID-Connect with a CA Certificate

To configure OpenID-Connect with a CA certificate, follow these steps:

Acquire your CA certificate (Review the documentation for your certificate provider for instructions) and store it in a secret by using the following example.

```
oc create secret generic <name of your kubernetes OIDC CA cert> --from-file=<path to your OIDC
CA cert file>
```

Modify the CR to identify the secret created.

```
oidcCaCertSecret: <name of your kubernetes OIDC CA cert>
```

Using your own images

If you built your own custom application images and want to deploy them, you can specify the image names in the CR by using the following example.

```
applicationDomain: miqproject.apps-crc.testing
orchestratorImage: docker.io/<your_username_or_organization>/<your_prefix>-orchestrator:latest
baseWorkerImage: docker.io/<your_username_or_organization>/<your_prefix>-base-worker:latest
uiWorkerImage: docker.io/<your_username_or_organization>/<your_prefix>-ui-worker:latest
webserverWorkerImage: docker.io/<your_username_or_organization>/<your_prefix>-webserver-
worker:latest
```

If you built your own operator image, you need to edit the operator deployment to specify the image.

Uninstalling

Uninstall IBM Cloud Pak® for AIOps, Infrastructure Automation, or both.

- Uninstalling IBM Cloud Pak for AIOps
- Uninstalling Infrastructure Automation

Uninstalling IBM Cloud Pak for AIOps

These instructions can be used to uninstall an online or offline (airgap) deployment of IBM Cloud Pak® for AIOps, and to clean up the resources that it created.

1. Run the uninstall script
2. (Optional) Delete the IBM Cloud Pak for AIOps catalog

Before you begin

The Red Hat® OpenShift® Container Platform command line interface (**`oc`**) must be installed on your Red Hat OpenShift cluster, and you must be logged in with **`oc login`**.

1. Run the uninstall script

Download the uninstall and cleanup script from here.

You can configure the script to uninstall and clean up your IBM Cloud Pak for AIOps deployment. Details on how to configure the script are provided in the same location.

You can configure which components and resources that were created by the IBM Cloud Pak for AIOps deployment are removed by the script. Review the following, and configure the script to remove them only if you are sure that they are not used by other CloudPaks on your Red Hat OpenShift cluster, and that you do not want to persist the data in them.

- Persistent Volume Claims (PVCs)
- shared components that are used by other Cloud Paks

Warning: Follow this step with caution if other IBM Cloud Paks® are installed on your Red Hat OpenShift cluster that might be sharing components and resources with your IBM Cloud Pak for AIOps deployment.

2. (Optional) Delete the IBM Cloud Pak for AIOps catalog

Warning: IBM Cloud Pak for AIOps is part of the **`ibm-operator-catalog`** catalog. Do not delete this catalog if you are using other operators from it.

If you do not require the **`ibm-operator-catalog`** catalog, then remove it with one of the following methods:

Online deployments:

- Option 1: Delete the IBM Cloud Pak for AIOps catalog with the Red Hat OpenShift console
- Option 2: Delete the IBM Cloud Pak for AIOps catalog with the Red Hat OpenShift CLI

Air-gapped (offline) deployments:

- Option 3: Delete the IBM Cloud Pak for AIOps catalog in an air-gapped deployment

Option 1: Delete the IBM Cloud Pak for AIOps catalog with the Red Hat OpenShift console

1. In the OpenShift console, select **Administration** > **Cluster Settings**.
2. Select the **Configuration** tab, and then select **OperatorHub** from the **Configuration Resource** column.
3. Select the **Sources** tab.
4. Select the CatalogSource that you want to remove.
5. Select **Delete CatalogSource** from the **Actions** list (the three vertical dots at the end of the row).

6. Verify that the CatalogSource that you removed is not shown in **Administration** > **Cluster Settings** > **Configuration** > **OperatorHub**.

Option 2: Delete the IBM Cloud Pak for AIOps catalog with the Red Hat OpenShift CLI

1. Run the following command for the catalog that you want to remove.

   ```
   oc delete catalogsource -n openshift-marketplace <catalogsource>
   ```

   Where **<catalogsource>** is the name of the CatalogSource that you want to delete.

2. Verify that the catalog that you removed is not returned by the following command:

   ```
   oc get catalogsource -n openshift-marketplace
   ```

Option 3: Delete the IBM Cloud Pak for AIOps catalog in an air-gapped deployment

Note: If you have an air-gapped deployment of IBM Cloud Pak for AIOps that was initially installed at a version lower than 3.5, then follow the instructions for **Option 3: Delete the AI Manager catalog in an upgraded air-gapped deployment** in the IBM Cloud Pak for AIOps 3.5 documentation: Uninstalling IBM Cloud Pak® for AIOps IBM Cloud Pak for AIOps.

If you have an air-gapped deployment of IBM Cloud Pak for AIOps, then remove the catalog by using the **uninstall-catalog** case action.

1. Export the following environment variables.

   The local Docker registry that you specify must be the registry in your air-gapped environment that you created when you installed Cloud Pak for AIOps.

   ```
   export CASE_NAME=ibm-cp-waiops
   export CASE_VERSION=1.8.0
   export CASE_INVENTORY_SETUP=cpwaiopsSetup
   export TARGET_REGISTRY_HOST=<IP_or_FQDN_of_target_registry>
   export TARGET_REGISTRY_PORT=<port_number_of_target_registry>
   export TARGET_REGISTRY=$TARGET_REGISTRY_HOST:$TARGET_REGISTRY_PORT
   ```

2. Run the **uninstall-catalog** CASE action.

   ```
   oc ibm-pak launch $CASE_NAME \
   --version $CASE_VERSION \
   --action uninstall-catalog \
   --inventory $CASE_INVENTORY_SETUP \
   --namespace openshift-marketplace \
   --args "--registry $TARGET_REGISTRY --recursive \
   --inputDir ~/.ibm-pak/data/cases/$CASE_NAME/$CASE_VERSION"
   ```

3. Verify that the catalog that you removed is not returned by the following command:

   ```
   oc get catalogsource -n openshift-marketplace
   ```

Note: If you are using IBM Storage Fusion or IBM Spectrum Scale Container Native storage, and some IBM® Automation foundation components remain after you complete uninstallation, then see the troubleshooting guidance in the IBM Cloud Pak foundational services documentation.

If the uninstallation fails, or is not complete and is not progressing, then see Troubleshooting installation and upgrade to help you identify any installation problems.

Uninstalling Infrastructure Automation

1. If you deployed Infrastructure Automation - Infrastructure management, delete the deployed *IMInstall* instance.

   1. Log in to your Red Hat OpenShift Container Platform cluster's console.
   2. From the main navigation menu, click **Operators** > **Installed Operators**.

3. Change the project to the project (namespace) where you installed Infrastructure Automation. For instance, *cp4aiops*.
4. Select the **IBM Management Infrastructure install** tab. The *IMInstall* instance is displayed.
5. Click the **IMInstall** under provided APIs column
6. Select **Delete IMInstall** in the Node Options menu for the IMInstall instance.
7. Wait for the *IMInstall* instance to be deleted.

2. Uninstall ManageService instance from IBM Managed services on the Red Hat OpenShift Container Platform console.

    1. Log in to your Red Hat OpenShift Container Platform cluster's console.
    2. Click **Operators > Installed Operators.**
    3. Change the project to the project (namespace) where you installed Infrastructure Automation. For instance, *cp4aiops*.
    4. Select the **IBM Management Infrastructure managed services** tab. The *ManageService* instance is displayed.
    5. Click the **ManageService** under provided APIs column
    6. Select **Delete ManageService** in the Node Options menu for the ManageService instance.
    7. Wait for the *ManageService* instance to be deleted.

3. Uninstall the *IAConfig* instance from the Infrastructure Automation operator.

    1. Log in to your Red Hat OpenShift Container Platform cluster's console.

    2. Click **Operators > Installed Operators.**

    3. Change the project to the project (namespace) where you installed Infrastructure Automation. For instance, *cp4aiops*.

    4. Select the **IBM Infrastructure Automation** tab. The **IBM Infrastructure Automation** instance is displayed. Click **IBM Infrastructure Automation** in the **provided APIs** column.

    5. Select **Delete IAConfig** in the Node Options menu for the IAConfig instance.

    6. Wait for the Managed services and Infrastructure management non-operator pods to be removed. To view the remaining pods, run the following command:

        ```
        oc -n <namespace> get pods
        ```

        Where **<namespace>** is the project (namespace) where Infrastructure Automation is installed.

        - Managed services pods are prefixed with **cam**.
        - Infrastructure management pods can be prefixed with any of the following prefixes: **1-**, **httpd**, **memcached**, **orchestrator**, **postgresql**, **ibm-infra-management**.

        Make sure that all the Managed services and Infrastructure management pods are removed before you move on to the next step.

4. Uninstall the Infrastructure Automation operators.

    1. Log in to the Red Hat OpenShift Container Platform console for your environment.
    2. From the main navigation menu, click **Operators** > **Installed Operators**.
    3. Change the project to the project (namespace) where you installed Infrastructure Automation. For instance, *cp4aiops*.
    4. Select **Uninstall Operator** from the options menu for the following operators:
        - **IBM Management Infrastructure install**
        - **IBM Management Infrastructure managed services**
        - **IBM Infrastructure Automation**

5. If you installed only Infrastructure Automation and not IBM Cloud Pak for AIOps in the Infrastructure Automation namespace, then uninstall the following operators in the installed namespace.

    1. Log in to the Red Hat OpenShift Container Platform console for your environment.
    2. From the main navigation menu, click **Operators** > **Installed Operators**.
    3. Change the project to the project (namespace) where you installed Infrastructure Automation. For instance, *cp4aiops*.
    4. Select **Uninstall Operator** from the options menu for the **IBM CloudPak functional services** operator.

6. Delete the namespace that Infrastructure Automation is deployed in.

    1. Run the following command to remove finalizers from **clients.oidc.security.ibm.com**.

```
oc -n <namespace> edit clients.oidc.security.ibm.com
```

Where **<namespace>** is the namespace that Infrastructure Automation is deployed in.

Delete the following finalizer lines, and save.

```
finalizers:
  - fynalyzer.client.oidc.security.ibm.com
```

2. Run the following command to remove finalizers from **operandrequests.operator.ibm.com**.

```
oc -n <namespace> edit operandrequests.operator.ibm.com
```

Where **<namespace>** is the namespace that Infrastructure Automation is deployed in.

Delete all the finalizer lines, and save.

```
finalizers:
   - finalizer.request.ibm.com
```

3. Delete the namespace that Infrastructure Automation is deployed in.

```
oc delete project <namespace>
```

Where **<namespace>** is the namespace that Infrastructure Automation is deployed in.

7. (Optional): Uninstall IBM Cloud Pak® foundational services

If you installed only Infrastructure Automation and not IBM Cloud Pak for AIOps in the Infrastructure Automation namespace, then use the following instructions in the IBM Cloud Pak foundational services documentation to uninstall IBM Cloud Pak foundational services.

Uninstalling foundational services

Defining connections and integrations

You can use the IBM Cloud Pak Automation console to define connections and applications.

The connections and applications that you create serve as the basis for:

- Incoming information that is used to model your IT infrastructure
- Observers that monitor your virtual environments
- Outgoing information that notifies you of any issues

For more information about defining different types of connections in IBM Cloud Pak for AIOps, see the following topics:

- Planning data loading, checking and training
- Mapping data
- Configuring data and tool connections
- Defining observer jobs

Netcool integrations

IBM Cloud Pak for AIOps can integrate with IBM Tivoli Netcool/OMNIbus and IBM Tivoli Netcool/Impact to leverage existing event data (IBM Tivoli Netcool/OMNIbus), and policies (IBM Tivoli Netcool/Impact).

With these integrations in place IBM Cloud Pak for AIOps provides AI and visualization features in a modernized UI with an IBM Tivoli Netcool/OMNIbus integration providing additional data and an IBM Tivoli Netcool/Impact integration providing additional automation. You can continue to take advantage of all of your alerts and integrations for IBM Tivoli Netcool/OMNIbus and IBM Tivoli Netcool/Impact to obtain and send data.

Integrating with IBM Tivoli Netcool/OMNIbus

You can integrate IBM Cloud Pak for AIOps with an existing IBM Tivoli Netcool/OMNIbus deployment to send event data through the connection to IBM Cloud Pak for AIOps. With this integration, you can see IBM Tivoli Netcool/OMNIbus event data in the IBM Cloud Pak for AIOps Alert Viewer. These events are converted into alerts on the IBM Cloud Pak for AIOps side.

You can then take advantage of the IBM Cloud Pak for AIOps AI features to manage and visualize the alerts. For example, event data can be correlated into incidents, data can be viewed and analyzed in a topology, and runbooks can be used to automate operations for managing the alerts.

To integrate IBM Cloud Pak for AIOps with IBM Tivoli Netcool/OMNIbus, you need to create a Netcool connection with the IBM Cloud Pak for AIOps UI Data and tools connections. This connection defines the settings for communication from IBM Cloud Pak for AIOps to IBM Tivoli Netcool/OMNIbus. For more information about how to define this connection, see Creating Netcool connections.

After you integrate with IBM Tivoli Netcool/OMNIbus, use IBM Cloud Pak for AIOps and the IBM Cloud Pak for AIOps UI for AI and visualization features. Use IBM Tivoli Netcool/OMNIbus and the associated UI tools for obtaining and sending data to IBM Cloud Pak for AIOps.

Note: If you have integrated IBM Cloud Pak for AIOps and IBM Tivoli Netcool/OMNIbus, and you are not yet ready to move to using the IBM Cloud Pak for AIOps UI, you can continue to use the IBM Tivoli Netcool/OMNIbus DASH custom dashboards and Web GUI event viewer as you transition to using the IBM Cloud Pak for AIOps UI. If you need to continue to use IBM Tivoli Netcool/OMNIbus tools, you can configure launch-in-context links from the Web GUI Event Viewer and the main DASH navigation to the IBM Cloud Pak for AIOps UI to help users access and use the IBM Cloud Pak for AIOps UI.

Integrating with IBM Tivoli Netcool/Impact

You can integrate IBM Cloud Pak for AIOps with an existing IBM Tivoli Netcool/Impact deployment so that IBM Cloud Pak for AIOps can then run automation policies that you manage with IBM Tivoli Netcool/Impact. New and existing policies can continue to be used to run operations in IBM Tivoli Netcool/Impact for data retrieval, drive automation, create tickets, leverage other integrations with IBM Tivoli Netcool/Impact, and more. These policies can be invoked by IBM Cloud Pak for AIOps or send events to IBM Cloud Pak for AIOps.

IBM Tivoli Netcool/Impact policies can reference the extension framework API for IBM Cloud Pak for AIOps. With these policies, you can write automation processes in IBM Tivoli Netcool/Impact to complete operations in IBM Cloud Pak for AIOps, such as to query alerts and incidents, and request changes to alerts.

To create this integration between IBM Cloud Pak for AIOps and IBM Tivoli Netcool/Impact, you need to first create an IBM Tivoli Netcool/Impact connection with the IBM Cloud Pak for AIOps UI Data and tools connections page. This connection defines the settings for communication from IBM Cloud Pak for AIOps to IBM Tivoli Netcool/Impact. For more information about how to define this connection, see Creating IBM Tivoli Netcool/Impact connections.

Then to complete the connection, you need to edit the RESTful DSA configuration in IBM Tivoli Netcool/Impact. This configuration defines the settings for communicating from IBM Tivoli Netcool/Impact to IBM Cloud Pak for AIOps. For more information about editing the RESTful DSA configuration, see Configuring a connection in IBM Tivoli Netcool/Impact for IBM Cloud Pak for AIOps.

Configuring data and tool connections

Connections enable IBM Cloud Pak for AIOps to collect data from various sources, then communicate how your IT operations are performing against a baseline model that is established by your AI models. To have a fully functioning instance of IBM Cloud Pak for AIOps, you must configure both outgoing and incoming data and tool connections. It is worth noting that ServiceNow is bidirectional.

For more information about connections in general, click the following links:

- Connections UI
- Connection types

For more information about connections in general, including at a minimum how to create, edit, and delete a particular connection, click the corresponding link in the following list:

- Ansible connections
- Appdynamics connections
- AWS CloudWatch connections
- Custom connections
- Dynatrace connections
- ELK connections
- Falcon LogScale connections
- Instana connections
- Kafka connections
- LogDNA connections
- Microsoft Teams connection
- Netcool connections
- New Relic connections
- Observer jobs
- Optional connections
- PagerDuty connections
- Slack connection
- ServiceNow connection
- Splunk connections
- SSH connections
- Zabbix connections

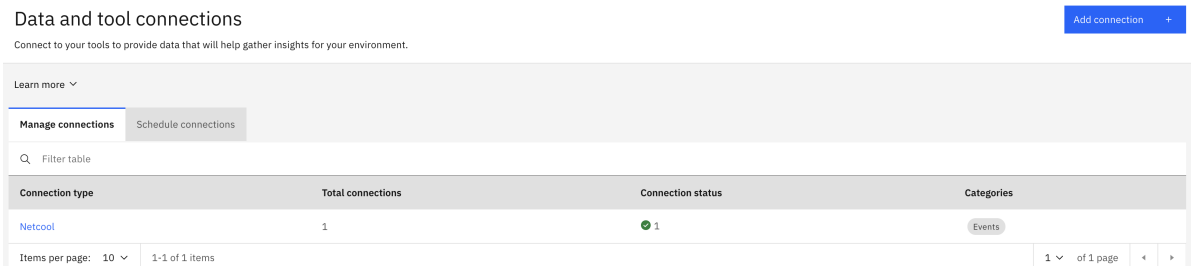
For ancillary information on connections, see:

- Event connections with probes and gateways
- Prerequisites for loading data from WebSphere log data sources
- Certificates

Data and tool connections

Using the Data and tool connections capability you can connect data sources to provide data that helps gather insights for your environment.

The main Data and tool connections page now has two tabs: the *Manage connections* tab, and the *Schedule connections* tab.



Data and tool connections UI

On the *Manage connections* tab, you can add connections, and view a list of links to previously added connection types, which when clicked, provide further information. On the *Manage connections* tab, clicking the options menu (three vertical dots) of a connection allows you to 'edit' or 'delete' that particular connection.

On the *Schedule connections* tab, you can manage and view your scheduled connections, and also to 'edit the schedule' of a connection, by using the options menu (three vertical dots) of that particular connection.

On the Data and tool connections *All connections* page, you can search and sort connections based on category, by using the search bar. You use the catalog page to create integrations.

Add connections

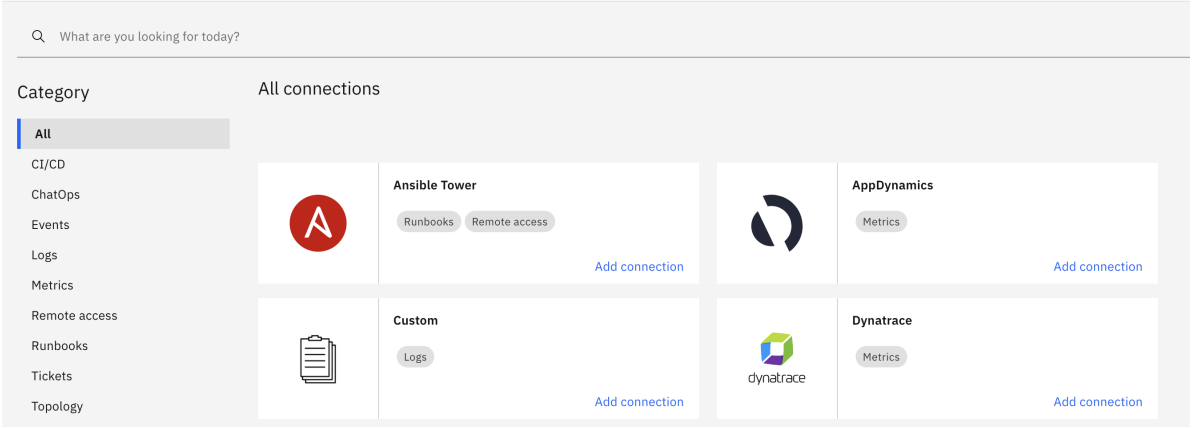


Figure. All connections page

You are aided by the relevant side-panel that pops up for each of the connection types you select, giving detailed information on what is needed to complete the forms.

Incoming connections provide data to IBM Cloud Pak for AIOps. Outgoing connections report findings from IBM Cloud Pak for AIOps to supported ChatOps interfaces and ServiceNow. Some connections are bidirectional. For example, to have a successfully operating instance of IBM Cloud Pak for AIOps, you might configure a Slack outgoing connection for ChatOps, a Falcon Logscsale incoming connection for log data, and a ServiceNow connection for change risk and similar ticket data.

Outgoing connections must be publicly accessible. If your cluster is behind a firewall, for example, it must be tunneled through something, such as Secure Tunnel. For more information about Secure Tunnel, see Using Secure Tunnel for ChatOps. If you previously exposed your host for one outgoing connection, you do not need to configure it for other outgoing connections.

For more information about prerequisites for specific outgoing connections, see the following topics:

- Configuring Microsoft Teams for connection
- Configuring Slack applications for connection
- Configuring the ServiceNow App for IBM Cloud Pak for AIOps
- Mapping data

Note: When you configure connections for IBM Cloud Pak for AIOps, your browser might give you an option to save key, token, or password-type fields. For security purposes, do not save these values. Turn off autofill or autosave settings in your browser, or exercise caution with these features.

Connection types

Use this information to understand which data connections to use to ingest different types of content.

Click the respective connection type that you want to learn more about creating, editing, and deleting that particular type of connection.

Connection types

Data source	Type of content supported	Link
Ansible	Runbooks, Remote Access	Ansible connections
Appdynamics	Metrics	Appdynamics connections
AWS CloudWatch connections	Metrics	AWS CloudWatch connections
Custom	Logs	Custom connections
Dynatrace	Metrics	Dynatrace connections
ELK	Logs	ELK connections
Falcon LogScale	Logs	Falcon LogScale connections
Instana	Events, Metrics, Topology	Instana connections

Data source	Type of content supported	Link
Kafka	Logs, Events	Kafka connections
LogDNA	Logs	LogDNA connections
Microsoft Teams	ChatOps	Microsoft Teams connections
Netcool	Events	Netcool connections
New Relic	Metrics	New Relic connections
PagerDuty	Events	PagerDuty connections
ServiceNow	Change tickets, Incident tickets	ServiceNow connection
Slack	ChatOps	Slack connections
Splunk	Logs	Splunk connections
SSH	Runbooks, Remote Access	SSH connections
Zabbix	Metrics	Zabbix connections

Outgoing connections

Outgoing connections enable IBM Cloud Pak for AIOps to communicate how your IT operations are performing against a baseline model that is established by you during model training. IBM Cloud Pak for AIOps uses this comparison to report anomalous activity to you in near real time.

Outgoing connections must be publicly accessible. If your cluster is behind a firewall, for example, it must be tunneled through something, such as Secure Tunnel. For more information about Secure Tunnel, see Using Secure Tunnel for ChatOps. If you previously exposed your host for one outgoing connection, you do not need to configure it for other outgoing connections.

When you set up your outgoing ChatOps connection, you must configure two channels: a *proactive channel* that you want to use for notifications (from change risk, for example), and a *reactive channel* that you want to use for incidents and alerts.

Note: The ServiceNow connection type has optional incoming and outgoing facets.

For more information about configuring ServiceNow, Slack, and Microsoft Teams for your connections, see the following topics:

- Microsoft Teams connection
  - Configuring Microsoft Teams applications for connection
- ServiceNow connection
  - Configuring the ServiceNow App for IBM Cloud Pak for AIOps
- Slack connection
  - Configuring Slack applications for connection

Creating a Microsoft Teams connection

A Microsoft Teams connection lets you and your team use collaborative ChatOps features to receive incident notifications. You can only connect to one ChatOps tool at a time, so if you already configured another ChatOps connection (such as Slack), you can't connect to Microsoft Teams.

Prerequisites

Before you set up your Microsoft Teams connection, you must create your Azure Bot, and configure the Microsoft Teams application to communicate with IBM Cloud Pak for AIOps. For more information about configuring Microsoft Teams applications for IBM Cloud Pak for AIOps, see Configuring Microsoft Teams applications for connection.

Important: If you have an offline (air-gapped) deployment of IBM Cloud Pak for AIOps or an environment that uses a proxy server, then you cannot use the ChatOps Microsoft Teams connection. The use of a proxy with the ChatOps Microsoft Team connection is not supported.

Note: If the URL of the on-premises IBM Cloud Pak for AIOps instance can't be accessed by Microsoft Teams, you can use Secure Tunnel to expose the IBM Cloud Pak for AIOps instance to either a cluster (Red Hat OpenShift or IBM Kubernetes Service cluster) or a host machine (virtual machine or physical machine), which is accessible from the public network. Then, you can get a URL that can be accessed by Microsoft Teams. For more information about Secure Tunnel, see Using Secure Tunnel for ChatOps.

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating a Microsoft Teams connection

Note: To implement your Microsoft Teams connection, you need the Client Secret and ID information available either from Microsoft Teams or Microsoft Azure. Follow the steps that are outlined in Configuring Microsoft teams application for connection to acquire these variables.

To create a Microsoft Teams connection, complete these steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.
4. From the list of available connections, find and click the **Microsoft Teams** tile.

   Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Connect**.
6. On the *Add connection* page, define the general connection details:
   - **Name**: The display name of your connection.
   - **Description**: An optional description for the connection.
   - **Microsoft App ID**: Token for the bot that is associated with the app that is installed in a channel. You can copy this value from the *Microsoft APP ID* field on the Microsoft Teams **Settings** > **Configuration** page.
   - **APP Password**: Enter the Azure Bot client secret for the connection.
   - **Language ID**: Select your language from the drop-down list.
   - **Proactive channel**: The channel ID for the channel that you want to use for notifications (from change risk, for example). For more information about obtaining channel ID values, see Obtaining links to your channels.
   - **Reactive channel**: The channel ID for the channel that you want to use for incidents and alerts. For more information about obtaining channel ID values, see Obtaining links to your channels.

   You can test your connection by clicking **Test connection**.

   Note: Your secret must be fully registered on Microsoft Azure for the connection to pass. If you test your connection immediately after creating the secret, wait a couple of minutes, then try again.
7. Click **Done**.

You have created a Microsoft Teams connection in your instance. Unlike a log or event connection, you do not need to enable a ChatOps connection after creating it. Once you create the connection and set up Microsoft Teams, you can verify that everything is working by issuing test commands in Microsoft Teams. For more information about verifying your connection, see Enabling your bot and adding it to your channel.

To create more connections, see Configuring data and tool connections.

For more information about working with the insights provided by your connections, see ChatOps insight management.

Connecting the Azure Bot to IBM Cloud Pak for AIOps

Before you can finish implementing your Azure Bot in Microsoft Teams, you must create your connection at the instance level in IBM Cloud Pak for AIOps. Creating a connection in IBM Cloud Pak for AIOps provides the URL for the messaging endpoint. After this step, you can edit the Microsoft Teams connection to complete setting it up, but the URL for the messaging endpoint is required to complete your Azure Bot registration.

1. Create a Microsoft Teams connection.
2. From the IBM Cloud Pak for AIOps Automation console, click the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Microsoft Teams** connection type in the *Manage connections* tab. On the *Microsoft Teams* page, copy the request URL from the *Details* column. This URL is required to enable two-way communication between Microsoft Teams and IBM Cloud Pak for AIOps.

   Note: If you used Secure Tunnel to expose the IBM Cloud Pak for AIOps for Microsoft Teams integration, you need to replace the `request URL` with the application mapping URL.

   To get the application mapping URL, go to **Administration** > **Secure Tunnel** from the navigation menu (four horizontal bars). On the **Manage connections** tab, find the tunnel connection, and click the connection name. In the **Application mappings** page, copy the application mapping URL from the **IBM Cloud Pak for AIOps mapping URL** field. For more information, see Viewing Secure Tunnel.
4. Log back into your Microsoft Azure home page and navigate to the *Configuration* section of your Azure Bot resource.
5. Paste the request URL from the *Details* column of the IBM Cloud Pak for AIOps *Microsoft Teams instances* page into the *Messaging endpoint* field on the *Configuration* section of your Azure Bot resource in Microsoft Azure. Click **Save**.

Enabling your Azure Bot and adding it to your channel

After you set up your Microsoft Teams connection, you can download the Microsoft Teams Azure Bot through the instance interface in the IBM Cloud Pak Automation console.

Note: The same Azure Bot must be added to both your reactive and proactive channels.

To download the Azure Bot, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Microsoft Teams* tile, click **1 instance**.
4. On the *Microsoft Teams instances* page, click the options menu (three vertical dots), then click **Download bot app**. Save `IBM_Cloud_Pak_AIOps.zip` to an accessible local directory.
5. Log in to Microsoft Teams to upload your Azure Bot to your channel. Click **Apps** in the sidebar.
6. Click **Upload a custom app**. Select the compressed file that you downloaded. The IBM Cloud Pak for AIOps Azure Bot appears in the *Built by your org* section.
7. Click the IBM Cloud Pak for AIOps tile, then click **Add**.
8. The tile is on the *Apps* page. Click the IBM Cloud Pak for AIOps tile on the *Apps* page. The IBM Cloud Pak for AIOps details page opens.
9. Click the drop-down arrow next to *Open*, then click **Add to a team**. Select the team that you designated to be the appropriate channel to post to (either your reactive or proactive channel). Click **Set up a bot**.

You have edited your connection.

The channel that you set up opens and the Azure Bot is fully functional.

You can verify that everything is operating as expected by issuing the welcome command for the Azure Bot in Microsoft Teams:

- `@<bot_name> /waiops_welcome`: provides a welcome message about using the bot, where `bot_name` is the name of your Azure Bot.

Editing a Microsoft Teams connection

After you create your connection, you can edit the connection. For example, if you want to change which channel your proactive messages are routed to, you can edit it. To edit a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Microsoft Teams** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the Microsoft Teams connections page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.
5. Edit your connection. Click **Next** to go through the connection configurationn pages. Click **Save** when you are done editing.

Note: Updating a ChatOps connection with a new Microsoft Teams Azure bot can result in disconnected ChatOps incidents. Existing incidents that were posted by a different bot do not receive any more updates. You are recommended to use the same bot resource when possible.

Deleting a Microsoft Teams connection

If you no longer need your Microsoft Teams connection, you can delete the connection from the console.

To delete a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Microsoft Teams** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the Microsoft Teams connections page, click the options menu (three vertical dots) for the connection that you want to delete and click **Delete**.
5. Enter the name of the connection to confirm that you want to delete your connection. Then, click **Delete**.

Your connection is deleted.

Configuring Microsoft Teams application for connection

IBM Cloud Pak for AIOps supports a direct connection to your enterprise Microsoft Teams installation.

To complete your Microsoft Teams connection in IBM Cloud Pak for AIOps, you need to obtain these credentials:

- Client secret
- Microsoft App ID
- Proactive channel ID
- Reactive channel ID

Prerequisites

- A Microsoft Azure account with the ability to create an Azure Bot and client secrets for your bot
- A Microsoft Teams account that you can create channels and add applications to

For more information about completing your Microsoft Teams connection, see Creating a Microsoft Teams connection.

Configuring Microsoft Teams

For a Microsoft Teams application connection, you must

1. Create an Azure bot resource
2. Enable Teams connection
3. Acquire connection credentials

1. Create an Azure bot resource

1. Log in to your Microsoft Azure home page.
2. Click **Create a resource**.
3. In the **Create a resource** search box, enter *Azure Bot* and select it from the list of available resources.
4. On the **Azure Bot** page, click **Create**.
5. When the create page opens, give the bot a name in the *Bot handle* field. This handle is used for tracking in Azure.
6. Next, from the dropdown menus, select a valid **Subscription** and **Resource** group.
7. In *Type of app*, select **Multi Tenant**.
8. At the bottom of the page, click **Next: Tags >**.
9. On the next page, enter any tags that you want to add in the **Tags** tab or skip and click **Review+create** tab.
10. Your submitted configuration is now validated and you can click the **Create** button.

The Azure Bot resource is now created. If you return to the Microsoft Azure home page, you see that the new Azure Bot resource is in the *Recent resources* section.

After creating and validating the Azure Bot resource, a notification appears about the deployment status. Click the notification **Deployment in progress**... to view the status of the resource, then **Go to resource** from the notification to manage your new resource.

2. Enable Teams communication

Retrieve the Azure Bot resource by going to the Microsoft Azure home page and clicking the *Recent resources* section. Alternatively, find the resource by entering the Azure Bot handle of the resource into the search box. Then, complete these steps:

1. From the resource page, click **Settings**. Go to the *Configuration* section.
2. Click **Channels**.
3. In the *Add a featured channel* section, click the Microsoft Teams icon. The *Configure Microsoft Teams* page opens.
4. Click **Save**, then accept the Terms of Service and click **Agree**.

The Azure Bot is now ready to connect to Microsoft Teams but before you implement the Azure Bot you must obtain the remaining information for your Microsoft Teams connection in IBM Cloud Pak for AIOps.

3. Acquire connection credentials

Retrieve the credentials of the Azure Bot resource by going to the Microsoft Azure home page and clicking the *Recent resources* section. Alternatively, find the resource by entering the Azure Bot handle of the resource into the search box.

From the resource page, then click **Settings** to access each of the credentials

- *Microsoft App ID*: Click **Configuration** > **Microsoft App ID**
- *Client Secret*: Click **Configuration** > **Microsoft App ID**. Click **Manage** > **New client secret**. (**Note:** This information is viewable only once, so make sure you copy the actual secret and not the ID.)

Proactive & Reactive Channel IDs: The instructions for both are the same, just with two different channels:

1. Log in to Microsoft Teams.
2. Click the options icon (three horizontal dots) next to the channel name for the channel that you want to use for your connection (proactive or reactive). Click **Get link to the channel**.

3. The channel ID is not the entire URL provided by the preceding step. To extract the channel ID from the URL, grab the string between channel/ and the next /.

   Example:
   `https://teams.microsoft.com/l/channel/19%3a7b1d2sldfglkjsdfhg73402d63%40thread.tacv2/...`
   In this example, the channel ID would be: **`19%3a7b1d2sldfglkjsdfhg73402d63%40thread.tacv2`**.

Now that you have created your Azure Bot and noted down the credentials, you have the required information for Creating a Microsoft Teams connection.

Creating a Slack connection

A Slack connection allows you and your team to use collaborative ChatOps features for receiving incident alerts and status. You can only connect to one ChatOps tool at a time, so if you already configured another ChatOps connection (such as Microsoft Teams), you won't be able to connect to Slack.

On the *Manage connections* tab you can create, edit, and delete connections, whereas on the *Schedule connections* tab you can manage and view your scheduled connections, and also to 'edit the schedule' of a connection.

Before you set up your Slack connection, you must create your bot and configure Slack to communicate with IBM Cloud Pak for AIOps. For more information about configuring a Slack application, see Configuring Slack applications for connection.

Note: If the URL of the on-premises IBM Cloud Pak for AIOps instance can't be accessed by Slack, you can use Secure Tunnel to expose the IBM Cloud Pak for AIOps instance to either a cluster (Red Hat OpenShift or IBM Kubernetes Service cluster) or a host machine (virtual machine or physical machine), which is accessible from the public network. Then, you can get a URL that can be accessed by Slack. For more information about Secure Tunnel, see Using Secure Tunnel for ChatOps.

For more information about working with a Slack connection, see the following sections:

- Creating a Slack connection
- Editing a Slack connection
- Deleting a Slack connection

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating a Slack connection

To complete your Slack connection, you need the following information:

- Signing secret
- Bot User OAuth Access Token
- Proactive channel ID
- Reactive channel ID
- If you have an offline (air-gapped) deployment of IBM Cloud Pak for AIOps , then you must have the details of a proxy server that you have set up for IBM Cloud Pak for AIOps to connect to Slack.

Obtaining these values require you to have the following prerequisites:

- Slack
- API authentication to your Slack instance
- If you have an online deployment of IBM Cloud Pak for AIOps, network connectivity between the IBM Cloud Pak for AIOps instance and the Slack Enterprise edition instance

  Note: If a direct endpoint cannot be established, you can use Secure Tunnel to securely tunnel between the two. For more information about Secure Tunnel, see Using Secure Tunnel for ChatOps.

Important: You can only create one ChatOps connection (Microsoft Teams or Slack) per instance.

For more information about setting up and configuring Slack for connection, see Configuring Slack for connection.

To create a Slack connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.
4. From the list of available connections, find and click the **Slack** tile.

   Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Connect**.
6. On the *Add connection* page, define the general connection details:

   - **Name**: The display name of your connection. If you have any restrictive EgressNetworkPolicies in place, ensure that they are updated to allow for this outbound connection.
   - **Description**: An optional description for the connection.
   - **Signing secret**: The secret that is generated for the client. You can copy this value from the *App Credentials* section inside the *Basic Information* page of the Slack application. For more information about obtaining this value, see Configuring your Slack app. If you edit this field, the secret for the current connection is updated. Leave this field blank if you do not want to update the secret.
   - **Language ID**: A dropdown list offering various different languages options for the connection.
   - **Bot token**: Token for the bot that is associated with the app that is installed in a workspace. You can copy this value from the *Install App* section in the Slack application. For more information about obtaining this value, see Configuring your Slack app. If you edit this field, the secret for the current connection is updated. Leave this field blank if you do not want to update the secret.
   - **Proactive channel**: Enter a proactive channel for the connection. The channel ID for the channel that you want to use for notifications (from change risk, for example). For more information about obtaining channel ID values, see Obtaining links to your channels.
   - **Reactive channel**: Enter a reactive channel for the connection. The channel ID for the channel that you want to use for incidents and alerts. For more information about obtaining channel ID values, see Obtaining links to your channels.

   On the optional *Configure Proxy* page, toggle the *Enable Proxy* selection to on if you require a proxy server to connect to Slack, and enter the following information:

   - **Proxy URL**: The hostname of your proxy server
   - **Proxy Port**: The port number
   - **Proxy username**: The proxy server username
   - **Proxy password**: The proxy server password

   You can test your connection by clicking **Test connection to Slack.com**.
7. Click **Done**.

You create a Slack connection in your instance. Unlike log or event connection, you do not need to enable ChatOps connections after creating them.

To create more connections, see Configuring data and tool connections.

For more information about working with the insights provided by your connection, see ChatOps insight management.

Editing a Slack connection

After you create your connection, your can edit the connection. For example, if you want to change which channel your proactive messages are routed to, you can edit it. To edit a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Slack** connection type on the *Manage connections* tab of the *Data and tool connections* page.

4. On the Slack connections page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.

   Note: The connections status in the *Manage connections* tab might not display the status of your Slack connection. When you are editing a Slack connection, click the **Test connection** button to ensure that your credentials are correct.

   You can test your connection by clicking **Test connection**. You can also test your connection by going to the target channel in Slack and entering `/welcome`. IBM Cloud Pak for AIOps responds with a welcome message in Slack.

5. Edit your connection. Click **Next** to go through the connection configurationn pages. Click **Save** when you are done editing.

Your connection is now edited. For more information about deleting a connection, see Deleting a Slack connection.

Note: Updating a ChatOps connection with a new Slack app can result in disconnected ChatOps incidents. Existing incidents that were posted by a different app do not receive any more updates. You are recommended to use the same app resource when possible.

Deleting a Slack connection

If you no longer need your Slack connection, you can delete the connection from the console.

To delete a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Slack** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the Slack connections page, click the options menu (three vertical dots) for the connection that you want to delete and click **Delete**.
5. Enter the name of the connection to confirm that you want to delete your connection. Then, click **Delete**.

Your connection is deleted.

Configuring Slack applications for connection

IBM Cloud Pak® for AIOps supports a direct connection to your enterprise Slack installation.

Prerequisites

The following requirements are necessary to setting up a Slack connection:

- Slack
- API authentication to your Slack instance
- A valid certificate (not self-signed) for your publicly exposed Red Hat OpenShift Container Platform route
- If you have an offline (air-gapped) deployment of IBM Cloud Pak for AIOps , then you must have the details of a proxy server that you have set up for IBM Cloud Pak for AIOps to connect to Slack.
- If you have an online deployment of IBM Cloud Pak for AIOps, network connectivity between the IBM Cloud Pak for AIOps instance and the Slack Enterprise edition instance.

  Note: If a direct endpoint cannot be established, you can use Secure Tunnel to securely tunnel between the two. For more information about Secure Tunnel, see Using Secure Tunnel for ChatOps.

Creating your Slack app

Complete the following steps to configure your Slack connection.

Note: Be careful which workspace that you use. You must either be a workspace administrator – or wait for approval from the workspace administrator – when you interact with some of Slack's administrative settings.

Important: IBM Cloud Pak for AIOps requires two Slack channels, one for *proactive* notifications and one for *reactive* notifications. You must configure two channels, one for each purpose.

1. Identify the Slack channel that you want to connect:
    1. Create or use an existing Slack channel.
    2. Retrieve the **`channel ID`**. Right-click the channel name, then click **Copy Link** to get the web URL link. The last part of the link is the **`channel ID`**, starting with *C* for public channels or *G* for private channels.
2. Create a Slack app on https://api.slack.com/apps
    1. Click **Create New App**.
    2. Enter an App Name, and specify the workspace that you want to connect the app to, then click **Create App**.

Configuring your Slack app

1. After you create your Slack app from the Slack API page, click the name of the app that you created, then click **Features** > **App Manifest**.
2. Replace the contents of the manifest file (YAML) with the following (make sure you update the **`<app name>`** to your own app name) and save the changes.

```
_metadata:
   major_version: 1
   minor_version: 1
display_information:
   name: <app name>
features:
   bot_user:
         display_name: <app name>
         always_online: false
oauth_config:
   scopes:
      user:
         - channels:history
         - chat:write
         - groups:history
      bot:
         - app_mentions:read
         - channels:manage
         - channels:read
         - chat:write
         - files:write
         - groups:read
         - groups:write
         - users:read
         - users:read.email
         - commands
         - mpim:read
settings:
   org_deploy_enabled: false
```

This manifest adds the following bot token to your application:

- **`app_mentions:read`**
- **`channels:manage`**
- **`channels:read`**
- **`chat:write`**
- **`files:write`**

- `groups:read`
- `groups:write`
- `users:read`
- `users:read.email`
- `commands`

To validate your changes were saved, you can navigate to **Features** > **OAuth, & Permissions** and review the *Bot Scopes* section.

3. Go to the *Settings* > *Install App* page and go to the OAuth Tokens for Your Workspace section.

    1. Click **Install to Workspace**.
    2. Review the information and click **Allow**. Note that you must reinstall your application every time you change the scope.
    3. Record the Bot User OAuth Token found in the *OAuth Tokens for Your Workspace* section.
4. On the **Settings** > **Basic information** page, go to select *App Credentials* section and record the *Signing Secret* value.

5. Using your Slack desktop application, create and name two Slack channels under your Slack workspace:

    - Proactive channel (for example, `aiops-proactive`)
    - Reactive channel (for example, `aiops-reactive`)

    Retrieve the channel ID for both channels. Right-click the channel name, then click **Copy link** to get the web URL link. The last part of the link is the channel ID (Such as `C020EA7QWLS`). You need these values in following steps.

6. Log in to the IBM Cloud Pak for AIOps console.

7. From the main navigation menu (four horizontal bars in the upper-left), expand **Define**, and click **Data and tool connections**. On the *Data and tool connections* page, click **Add connection**.

8. Select *ChatOps* in the *Category* list (on the left), and click the **Slack** tile. On the side-panel, review the instructions and when ready to continue, click **Connect**.

9. Enter the following information on the Slack integration definition page:

    - `Name`: The display name of your integration (Such as Slack Chat Ops)
    - `Description`: An optional description for the integration.
    - `Signing secret`: The signing secret generated for the client. You can copy this value from the App Credentials section inside the *Basic Information* page of the Slack application.
    - `Bot token`: This is the token value for the bot that you recorded earlier. You can copy this value from the *Settings* > *Install App* page section in the Slack application.
    - `Language ID`: A dropdown list offering various different languages options for the connection.
    - `Proactive channel`: The channel ID for the channel that you want to use for notifications. You recorded this value in step 5.
    - `Reactive channel`: The channel ID for the channel that you want to use for incidents and alerts. You recorded this value in step 5.

    On the optional **Configure Proxy** page, toggle the *Enable Proxy* selection to on if you require a proxy server to connect to Slack, and enter the following information:

    - `Proxy URL`: The hostname of your proxy server
    - `Proxy Port`: The port number
    - `Proxy username`: The proxy server username
    - `Proxy password`: The proxy server password
10. Test your Slack integration connection by clicking **Test connection**. This test verifies that IBM Cloud Pak for AIOps can communicate with Slack (though it does_not_verify if Slack can communicate with IBM Cloud Pak for AIOps). If the test fails, review all preceding steps described previously.

11. Click **Done**.

12. From the Slack application page, go to *Features* > *App Manifest*. Complete the following updates to the contents of the manifest YAML.

    - Under the *settings* element, add the following child items. Ensure that you update the **placeholders**. The `<request URL>` value is the URL that you get from the console on the Data and tool connections. For example, `https://myibm.com/aiops/aimanager/instances/1647499820476511/api/slack/events`. You can also get the value from the secure tunnel section Using Secure Tunnel for ChatOps.

```
      event_subscriptions:
         request_url: <request url>
         bot_events:
            - app_mention
            - member_joined_channel
      interactivity:
            is_enabled: true
            request_url: <request url>
```

`event_subscriptions` and `interactivity` are at the same level in the configuration as the `org_deploy_enabled` and `socket_mode_enabled` elements.

- Under the `features` element, add the following child item. Ensure that you update the **placeholders**.

```
      slash_commands:
         - command: /welcome
            url: <request url>
            description: Welcome to IBM Cloud Pak for AIOps
            should_escape: false
```

`slash_commands` is at the same level in the configuration as the `bot_user` element.

13. Verify the Request URL. Click **Click here to verify link**. If this URL verification is not successful, review all preceding steps and ensure that the Request URL is correct.
14. Using the Red Hat OpenShift command line, run the following commands for the changes to take effect:

    ```
    oc project <namespace>
    ```

    Where `<namespace>` is the namespace where IBM Cloud Pak for AIOps is installed.

    ```
    oc set env deployment/$(oc get deploy -l app.kubernetes.io/component=chatops-slack-
    integrator -o jsonpath='{.items[*].metadata.name }') SLACK_WELCOME_COMMAND_NAME=/welcome
    ```

15. After the Slack application is configured, it must be reinstalled, as several permissions changed. Go to *Settings* > *Install App* page and click **Reinstall to Workspace**. On the subsequent page, click **Allow**.
16. Add the Slack application to **both** channels that you created earlier, such as your *proactive* and *reactive* channels.
    - Using your Slack desktop application, open the *Details* view of the channel, click **Integration** > **Add apps**, and add the Slack application to the channel. (The channel's *Details* view is found by clicking the small down-arrow next to the channel's name in the Slack UI.)
    - After you add the application to the channel, send the message `/welcome` to the channel. As a response, you can get a welcome message posted to the channel, which confirms the connection was made successfully.

You can now use Slack as your ChatOps interface. For more information about using the ChatOps interface, see ChatOps insight management.

Configuring the ServiceNow App for IBM Cloud Pak for AIOps

Before you set up your ServiceNow connection in IBM Cloud Pak® for AIOps, you must set up the IBM Cloud Pak for AIOps ServiceNow App in your ServiceNow instance.

About this task

With this connection, you can create *Incidents* in IBM Cloud Pak for AIOps that then become *Incidents* in ServiceNow. Both contain the same information. To set up this connection, you must set up ServiceNow and the ServiceNow connection in IBM Cloud Pak for AIOps.

When the connection is created, the ServiceNow App can receive updates from IBM Cloud Pak for AIOps for the state, priority, and description values. However, updates from the ServiceNow App are not sent back to IBM Cloud Pak for AIOps.

Prerequisites

Before you can set up ServiceNow to connect with IBM Cloud Pak for AIOps, you need to make sure that your ServiceNow instance meets the following requirements:

- Install the IBM Cloud Pak for AIOps App into your ServiceNow instance. To get the app, go to the ServiceNow Store.
- Your ServiceNow application must be version 4.2.1 to connect with IBM Cloud Pak® for AIOps version 4.2.0. This connection is not supported for earlier versions of the ServiceNow application.

Procedure

Complete the following steps to set up ServiceNow for connection:

1. Assign the managing roles for granting access to ServiceNow.
2. Create the users that need to access and set up the connection.
3. Create the ServiceNow connection with the IBM Cloud Pak for AIOps UI

Managing roles

In ServiceNow, two preset roles can be assigned to users: **`user_admin`** and **`itil`**. These roles can be added to the existing roles in the IBM Cloud Pak for AIOps App to grant more access to ServiceNow functions.

- The **`user_admin`** role gives privileges to create new users and assign roles to users.
    - Adding this role is optional. Not having this role does not affect functioning.
- The **`itil`** role gives access to the incident and change ticket views.
    - This role is required to view the IBM Cloud Pak for AIOps incidents and change risk assessments.

1. Log in to your ServiceNow custom application.
2. In the filter field, search for **`roles`**.
3. Click **User Administration** > **Roles**.
4. Search for **`x_ibm_waiops`**, which produces a list of predefined roles.

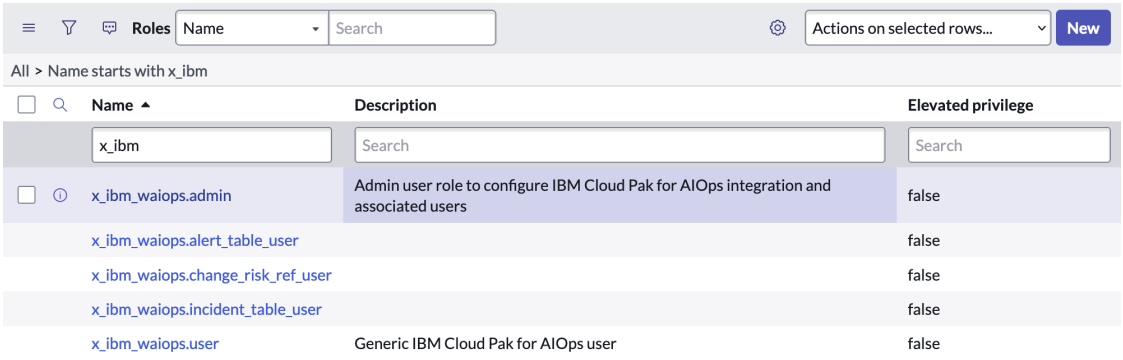


Figure. ServiceNow title list

5. Click **`x_ibm_waiops.admin`**.
6. **Optional:** Under the *Contains Roles List* box, select **Edit**, and add the **`user_admin`** role to the **`x_ibm_waiops.admin`** role.

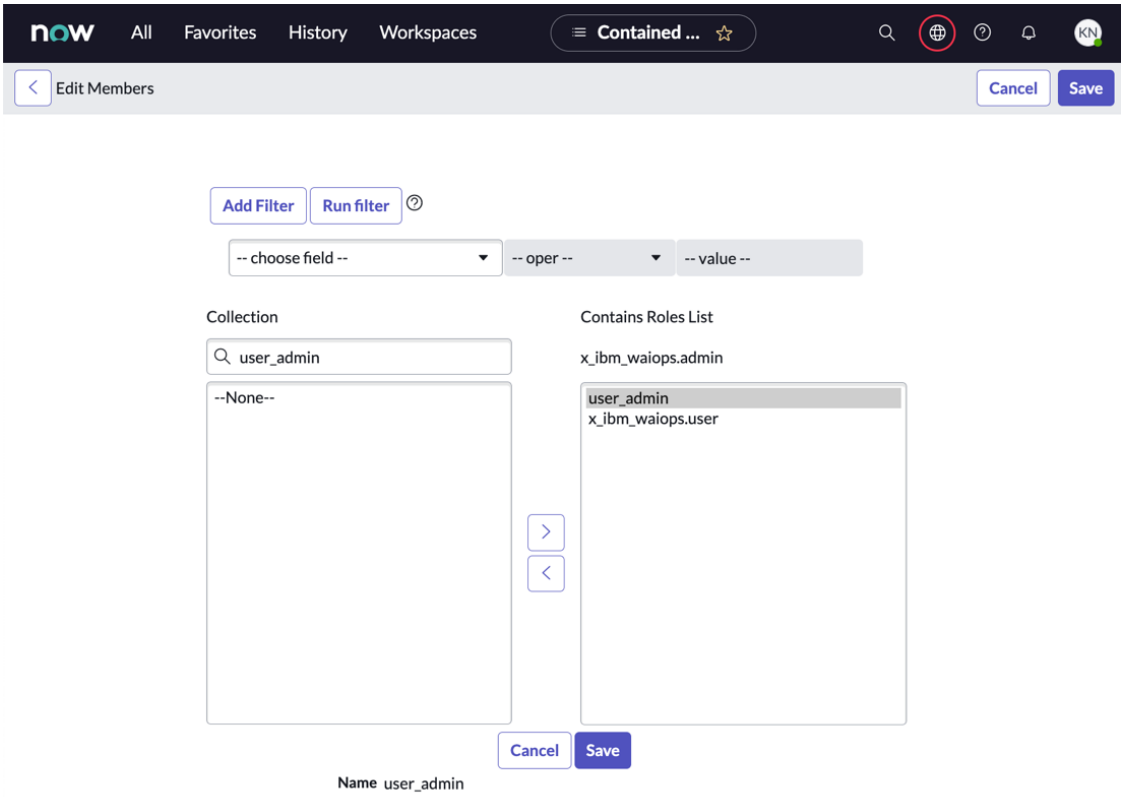


Figure. ServiceNow edit members

7. Go back to the list of predefined roles. Select **`x_ibm_waiops.user`**.
8. In the *Contains Roles List* box, select **Edit**, and add the **`itil`** role to **`x_ibm_waiops.user`**. Unlike the **`user_admin`** role, this role is required.

    - Make sure that your IBM Cloud Pak for AIOps users in ServiceNow are assigned either the **`x_ibm_waiops.admin`** or the **`x_ibm_waiops.user`** role. The **`x_ibm_waiops.admin`** role includes the **`x_ibm_waiops.user`** role, so it inherits the **`itil`** role. All IBM Cloud Pak for AIOps users need the **`itil`** role to view generic incidents and change tickets. Without either the **`x_ibm_waiops.admin`** or the **`x_ibm_waiops.user`** roles, a user cannot access the augmented incidents and change tickets from the IBM Cloud Pak for AIOps ServiceNow App and receives a permissions error.
    - The other three roles available within the **`x_ibm_waiops.user`** role (all of which begin with **`x_ibm_waiops.`**) provide access to the three new tables introduced by the IBM Cloud Pak for AIOps ServiceNow App.

9. Now you see something like this:

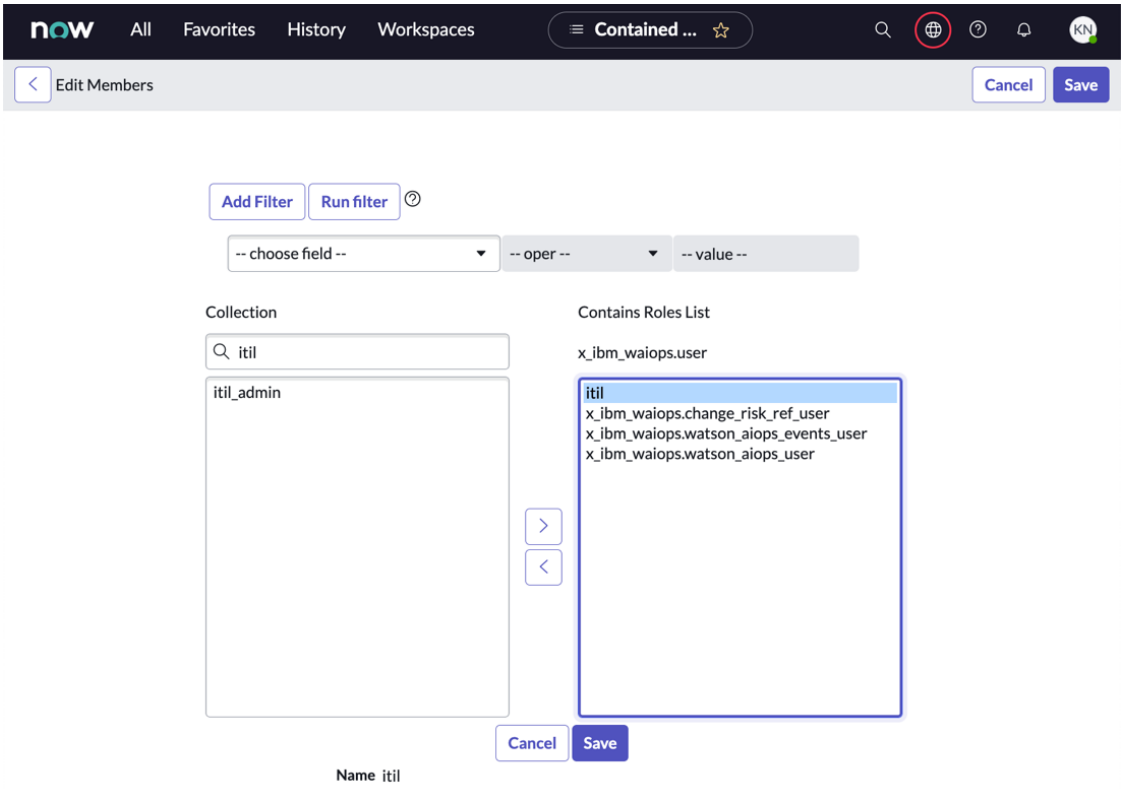


Figure. ServiceNow roles list

User roles

1. Go to **Organization** > **Users**. Create at least 2 users, an admin and a generic user to be a Site Reliability Engineer (SRE).
    - For the admin user, for the built-in filters to work, the admin's user ID needs to be **`ibm.aiops`**, and the admin's first name needs to be **`IBM Cloud Pak for AIOps`**.
    - For the SRE user, the user ID and first and last name do not need to be specific values.
2. For each user, set up a password by clicking **Set Password**. After you click **Set Password**, ServiceNow assigns a default password.
3. Copy and save the password. The first time that you log in with the default password, ServiceNow prompts you to set a new password. Make sure to change the default password.
4. For each user, find the **Roles** tab.

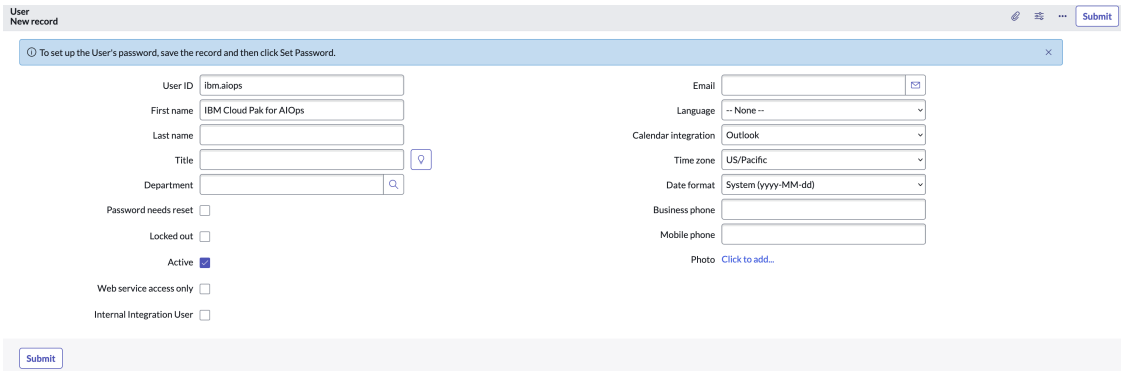


Figure. ServiceNow user creation

5. For the admin user, assign the **`x_ibm_waiops.admin`** role to be used in step 6 of Creating a ServiceNow connection.

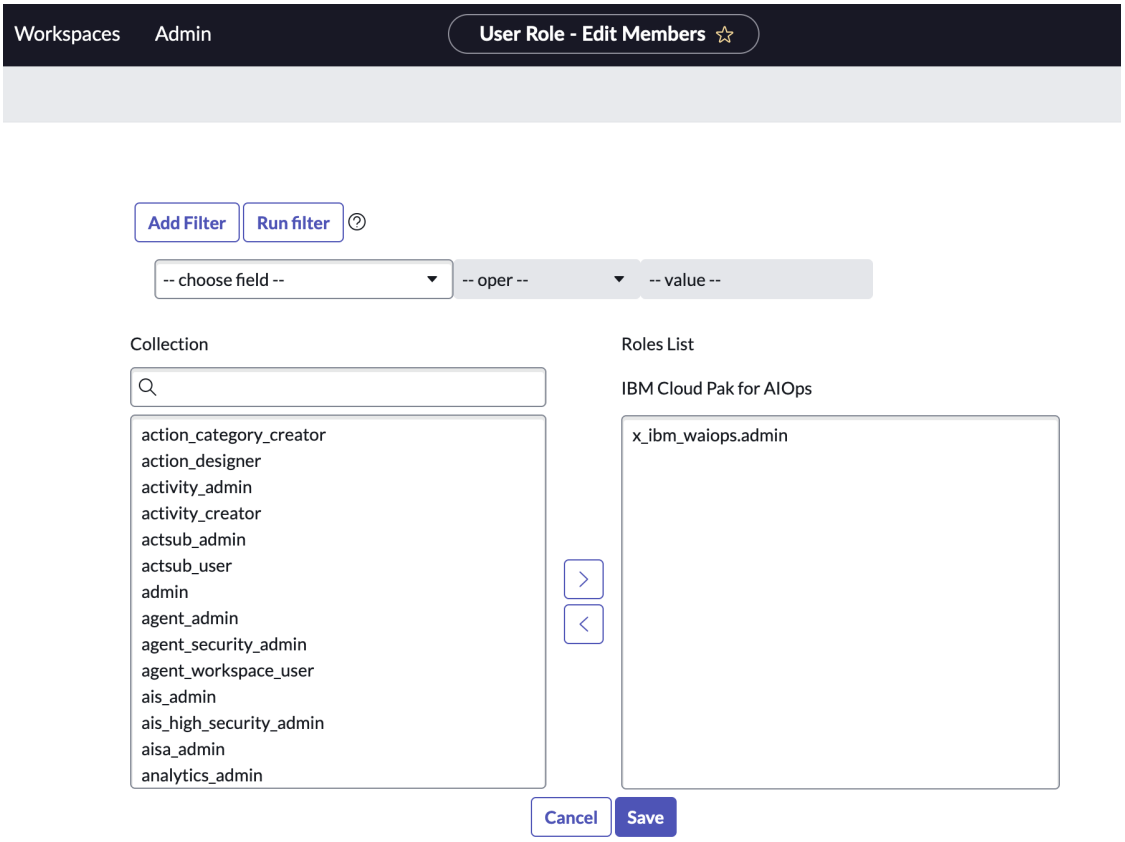


Figure. ServiceNow role assignment

This admin user has their name **`IBM Cloud Pak for AIOps`** attached to the generated incidents and change ticket assessments that are created by IBM Cloud Pak for AIOps in ServiceNow.

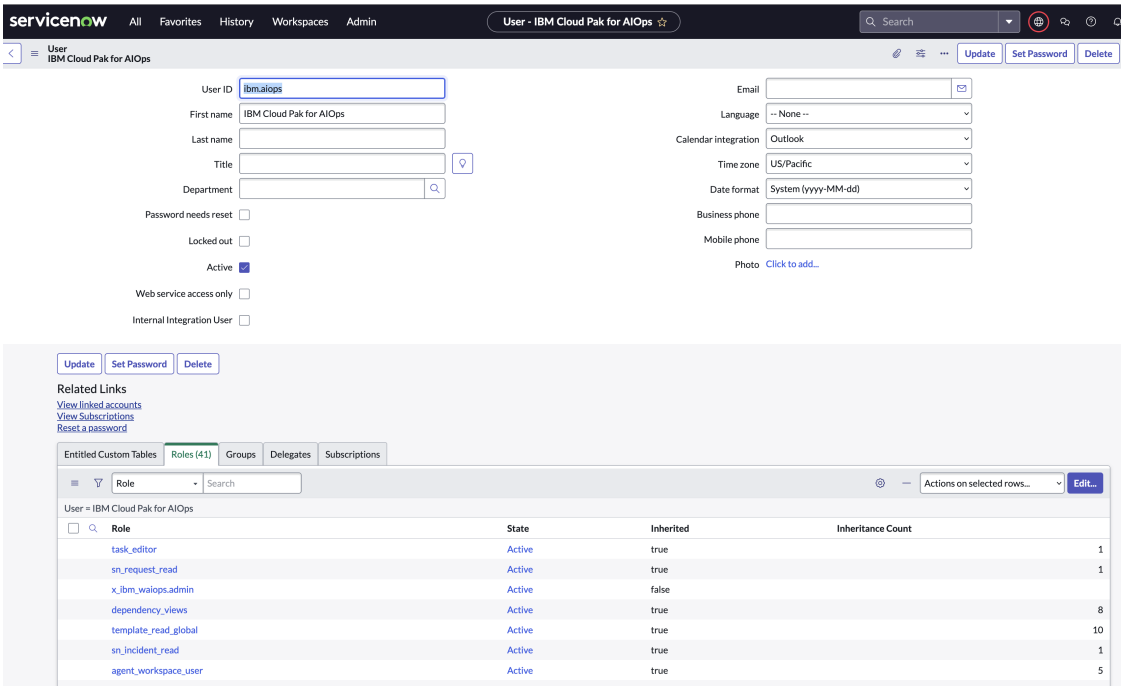


Figure. ServiceNow user roles

6. For the SRE user, assign the **`x_ibm_waiops.user`** role so that the user can view and assess ServiceNow incidents and change tickets.

Important:

- Make sure that the time zone of your *admin user* (the user value that is used to create your ServiceNow connection) matches your system time zone in ServiceNow. If the two values are not synchronized, the flow of your change request data from ServiceNow to IBM Cloud Pak for AIOps can be disrupted.
- To view the details provided by the ServiceNow connection, switch to the **IBM Cloud Pak for AIOps** view as shown in the following image:

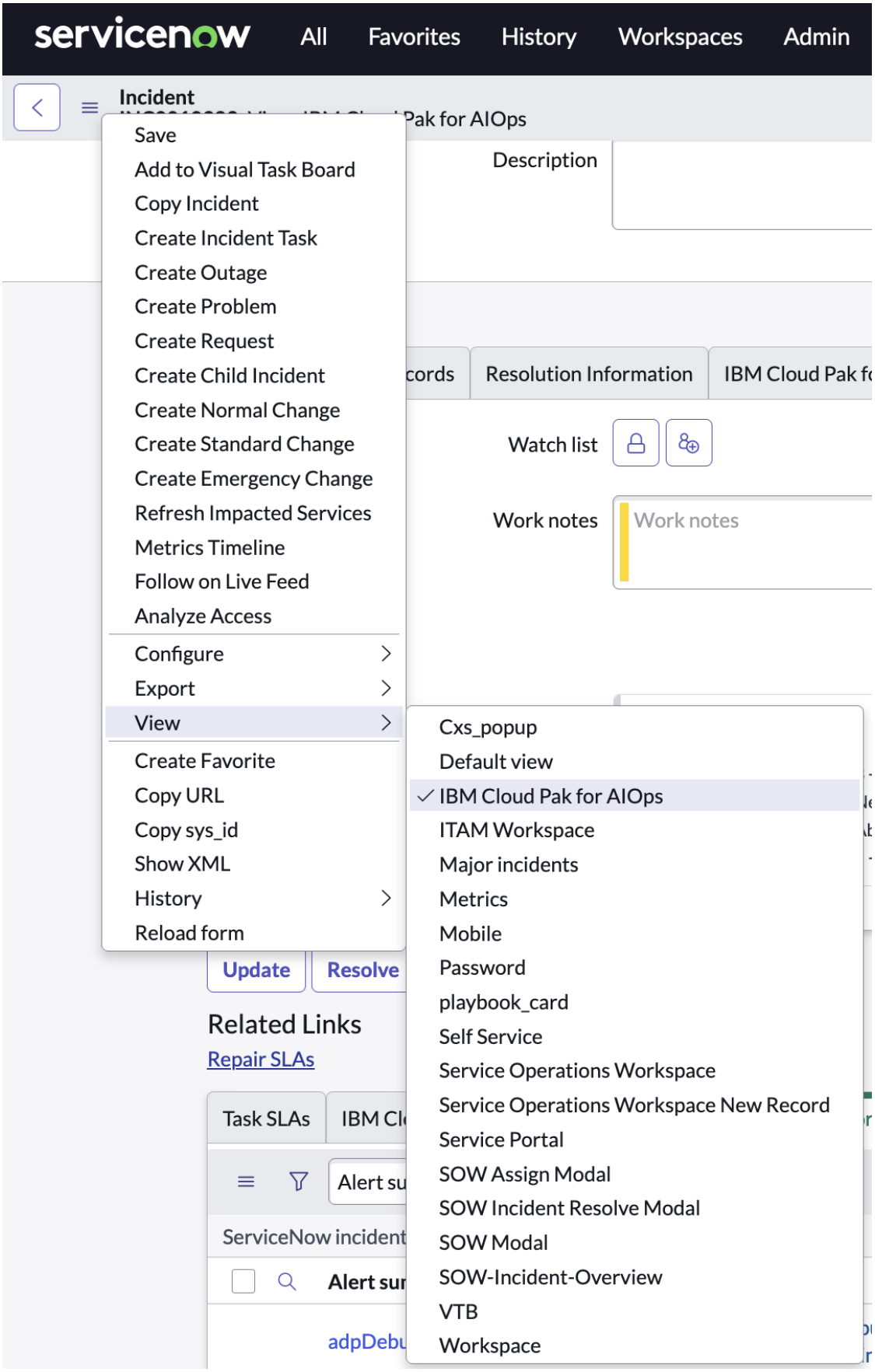


Figure. Cloud Pak View for ServiceNow connection

Create the ServiceNow connection

You also need the ServiceNow connection to be created with the IBM Cloud Pak for AIOps UI. For more information, see Creating a ServiceNow connection.

ServiceNow App configuration page

Although the **System properties for integration with IBM Cloud Pak for AIOps** page is still visible in the ServiceNow App, you do not need to configure it.

You can configure ServiceNow to communicate back to IBM Cloud Pak for AIOps by clicking **ALL RESULTS** and expanding **IBM Cloud Pak for AIOps** > **Configuration (Beta)** in the ServiceNow UI.

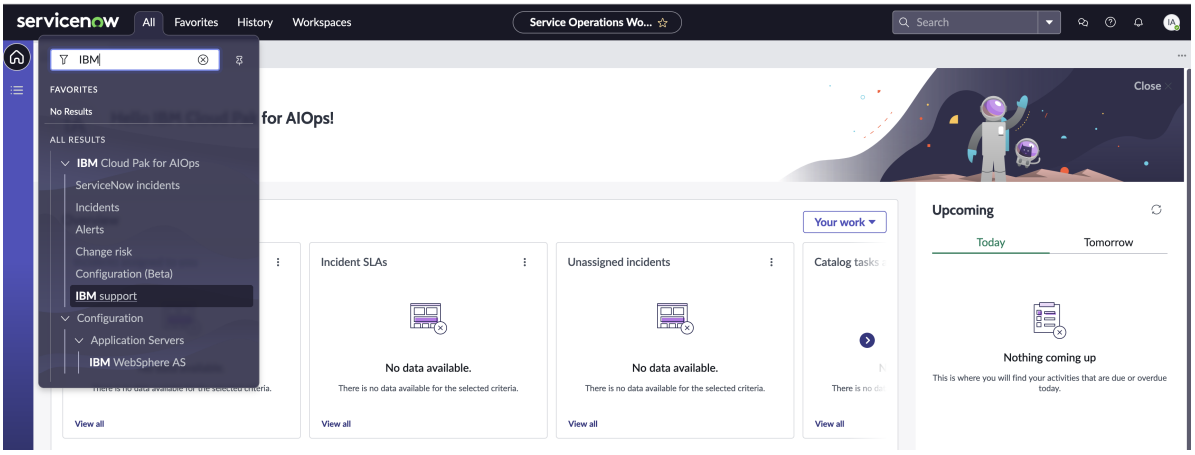


Figure. ServiceNow UI menu for IBM Cloud Pak for AIOps

Creating a Change Risk Assessment

Complete the following steps to create and view a change request.

1. Within the IBM Cloud Pak® for AIOps WebUI, make sure ServiceNow Integration is set to the *Live data for continuous ticket data collection* mode. For more information, see step 11 of Creating a ServiceNow connection.
2. You also need a ServiceNow connection and a trained and deployed Change Risk model. For more information, see Training a Change Risk model.
3. Within your ServiceNow instance, go to **Change** > **All** to view all existing change requests.
4. Click **New** > **Create Normal Change**. Add a short description for your change. Then, click **Submit**. If ServiceNow sends you to another page, select **Change** > **All** and find the change ticket that you created.

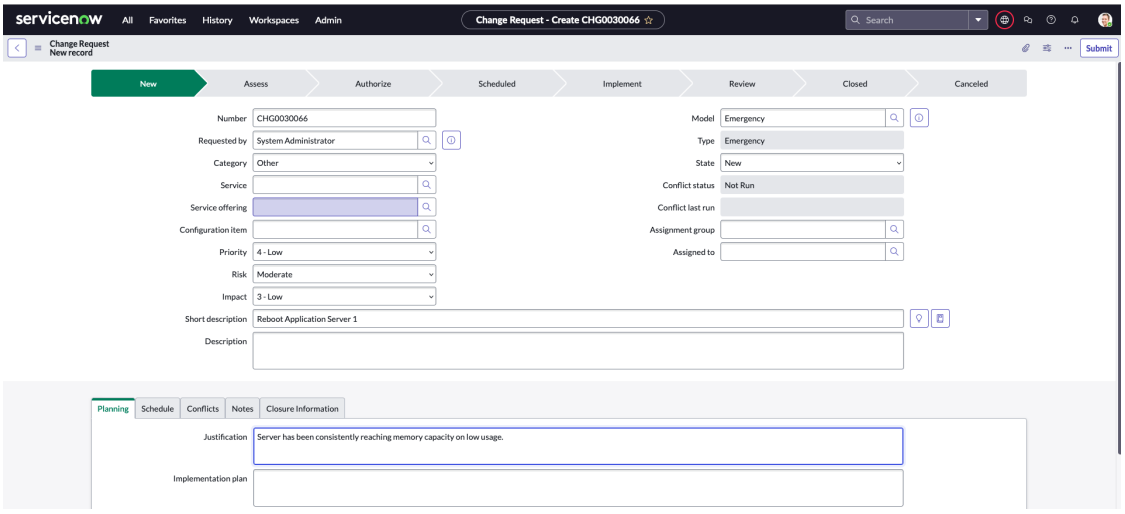


Figure. Change request short description

5. View the Change Request and the Change Risk Assessment tab.

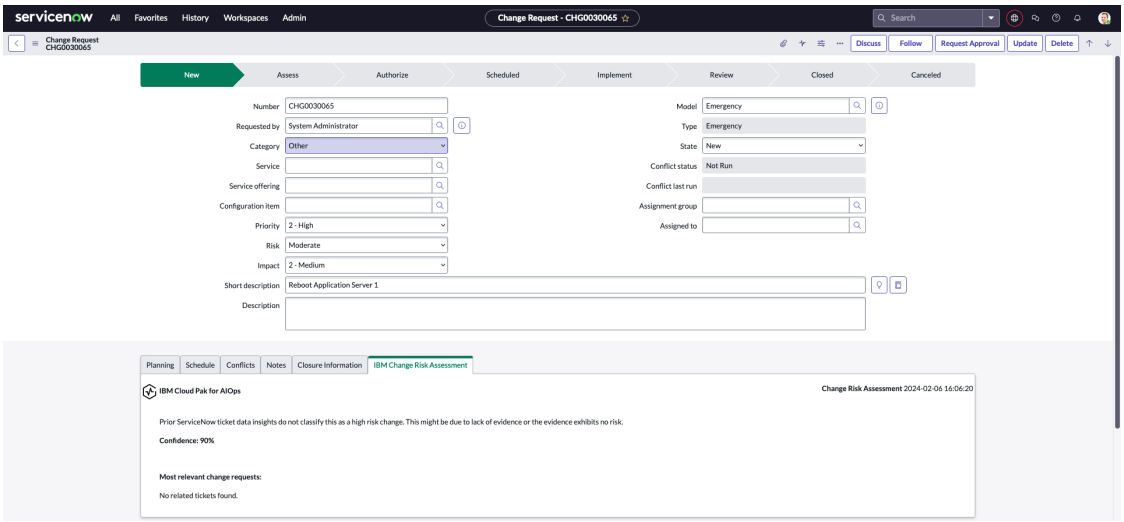


Figure. Change Request Assessment tab

Incoming connections

Incoming connections enable IBM Cloud Pak for AIOps to collect log, metric, and event data from various sources.

IBM Cloud Pak for AIOps uses this collected data to establish baseline models during AI model training. These models are then used to compare against live incoming data from these connections to report anomalous activity to you in near real time by way of your outgoing connections. In live data mode, the connector might need a delay to offset the query time windows to provide time buffer to prevent retrieving partial real-time data. For more information, see Delay configuration in data connections.

In the Connector UI, there is an environment variable that allows you to optionally configure how often your connection statuses are refreshed in the UI. If you do not modify this value, the UI will use the default of 300 seconds (5 minutes). To customize this value, see Optional caching configuration for Connector UI.

The recommended value for source parallelism should be equal to the number of days. For example, if you are pulling historical data for training for 7 days, the source parallelism should be around 7 or higher, so that you can pull data for multiple days simultaneously.

Unlike outgoing connections, you can set up multiple sources of incoming connections. For example, if you have a single source of Falcon LogScale data for logs, and PagerDuty configured for events, you can create connections for both of those. You can also use the custom and Kafka integrator for greater flexibility. The custom connection type collects data from a single specific data source (like any other connection type). If you want to collect log data from different systems, and then route that information through a forwarding agent, such as an Apache Kafka topic (like Sysdig or FluentBit), you can use the Kafka connection type.

Note: The ServiceNow connection type has optional incoming and outgoing facets.

For more information about incoming connections, see the following topics:

- Administrator process connections
- Ansible connections
- Appdynamics connections
- AWS CloudWatch connections
- Custom connections
- Dynatrace connections
- ELK connections
- Falcon LogScale connections
- Generic webhook connection
- IBM Tivoli Netcool/Impact connection

- Instana connections
- Kafka connections
- LogDNA connections
- Netcool connections
- New Relic connections
- Observer connections
- PagerDuty connections
- Splunk connections
- SSH connections
- Zabbix connections

Event and metric connections

In addition to the default data and tool connections that are provided by IBM Cloud Pak® for AIOps, you can configure optional connections to get events, topology, and metrics by using administrator processes.

The following optional connections can be customized by providing your own data and rule sets.

- Probes: Probes connect to an event source, detect, and acquire event data, and forward the data to the IBM Tivoli Netcool/OMNIbus ObjectServer as events. Probes use the logic that is specified in a rules file to manipulate the event elements before adding that data into the fields of an event in the ObjectServer alerts.status table.

  They can acquire data from any stable data source, including devices, databases, and log files. Some probes are uniquely designed to acquire event data from a specific source, while other more generic probes (for example, the Message Bus Probe) can be configured to integrate with various sources. The IBM® Netcool® Operations Insight® Event Integrations Operator can deploy multiple connections, for example: it can run instances of the Prometheus Probe, CEM Probe and CEM Gateway. For more information about IBM Tivoli Netcool/OMNIbus, see Operators.

- gateways: Gateways enable the exchange of events between ObjectServers and complementary third-party applications, such as databases, and helpdesk or Customer Relationship Management (CRM) systems. You can use gateways to replicate events or to maintain a backup ObjectServer. You can use application gateways to integrate different business functions. For example, you can configure a gateway to send event information to a helpdesk system. You can also use a gateway to archive events to a database. After a gateway is correctly installed and configured, the transfer of events is apparent to operators. For more information, see Install IBM Tivoli Netcool/OMNIbus.

  The IBM Netcool Operations Insight Event Integrations Operator can deploy multiple connections, for example, it can run instances of the Prometheus Probe, CEM Probe, and CEM Gateway. For more information about IBM Tivoli Netcool/OMNIbus, see Operators.

Ansible Automation Platform connections

You can create a connection to Ansbile Automation Platform. Red Hat Ansible Automation Platform provides tools for building and operating automation.

IBM supports Ansible as part of the Ansible Automation Platform. For more information about this platform, see Red Hat Ansible Automation Platform.

Note: You can create only one Ansible Automation Platform controller connection. Also, you must have an account with administrator role to create, edit, view, or delete an Ansible Automation Platform connection.

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating Ansible Automation Platform connections

Complete the following steps to create a connection to an Ansible Automation Platform controller server.

1. Log in to IBM Cloud Pak Automation console.

2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.
4. From the list of available connections, find and click the **Ansible Automation Controller** tile.

   Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Connect**.
6. Enter the base URL of your Ansible Automation Platform controller server. This URL must contain the protocol, for example: **`https://ansible.mycompany.com:443`**.
7. Choose an authentication type. You can select **User ID/Password** to connect with username and password, or **API Token** to use a bearer token, previously created with **Write Scope** in the Ansible.
8. Enter the chosen authentication information.
9. Optional: Enter the Ansible Automation Platform controller server root CA certificate, or the Ansible Automation Platform controller self-signed certificate.

   If the Ansible server certificate is based on a root CA certificate, then this root CA certificate must be entered (in PEM format). Some Ansible installations provide a self-generated root CA certificate (for example, at **`<aap_install_dir>/tls/ca.cert`**).

   If the Ansible server certificate is a self-signed certificate, then make sure the common name (CN) of the certificate matches the fully qualified domain name of the host system (a certificate with another CN like "localhost" will not be accepted). If needed, replace that certificate with a certificate that is issued for the actual domain name you are using.

   On Linux systems, enter the following command to receive the self-signed certificate:

   ```
   echo -n | openssl s_client -servername <ANSIBLE_TOWER_HOSTNAME> -connect
   <ANSIBLE_TOWER_HOSTNAME>:<ANSIBLE_TOWER_PORT> -showcerts | sed -ne '/-BEGIN
   CERTIFICATE-/,/-END CERTIFICATE-/p' > file.cert
   ```

   If the command does not work in your environment, use the following variant of the command:

   ```
   ex +'g/BEGIN CERTIFICATE/,/END CERTIFICATE/p' <(echo | openssl s_client -showcerts -
   servername <ANSIBLE_TOWER_HOSTNAME> -connect <ANSIBLE_TOWER_HOSTNAME>:
   <ANSIBLE_TOWER_PORT>) -scq > file.cert
   ```

   If errors occur, make sure your exported certificate that is stored in **`file.cert`** contains a full and valid certificate. Errors like **`verify error:num=20:unable to get local issuer certificate`** occur due to a missing CA root certificate for the DigiCert CA.

   The resulting file.cert might contain one or more certificates and each certificate begins and ends as follows:

   ```
   -----BEGIN CERTIFICATE-----
   ...
   -----END CERTIFICATE-----
   ```

   On Windows systems, use your preferred browser to export the certificate or certificate chain.
10. Click **Done** to complete the Ansible Automation Platform integration.

Note: See Ansible actions for additional information on the supported Ansible versions and some other restrictions.

Creating AppDynamics connections

AppDynamics connections collect different types of metrics, including live and historical data, and send them to the gRPC server as Cloud events at specified intervals.

You need to have both an endpoint and basic authentication ready. The AI Algorithm that analyzes AppDynamics data is Metric anomaly detection. AppDynamics SaaS controller 22.5 is the version supported for the AppDynamics Connector.

For more information about working with AppDynamics connections, see the following sections:

- Creating AppDynamics connections
- Enabling AppDynamics connections
- Editing AppDynamics connections
- Deleting AppDynamics connections
- Adding an AppDynamics data retention period

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating AppDynamics connections

To create an AppDynamics connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.
4. From the list of available connections, find and click the **AppDynamics** tile.

   Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Connect**.
6. On the *Add connection* page, define the general connection details:

   - **Name**: The display name of your connection. You cannot use spaces or special characters in the display name. Use only alphanumeric characters.
   - **Description**: An optional description for the connection.
   - **Client ID**: Client ID is generated from the AppDynamics controller, in the format 'ApiClientName@AccountName', for example 'aiops@ibm', where 'aiops' is the API Client name, and 'ibm' is the AppDynamics account name.
   - **Client Secret**: Generated from **AppDynamics UI -> Settings -> Administration -> API Clients**. Click **+** (create), and enter an API Client Name. Also, ensure that the Client Secret created has the following roles: Account Owner, Administrator, or Applications and Dashboards Viewer, which is the minimum role. If not, the API calls might not return valid application IDs.
   - **Endpoint**: Enter an endpoint for the connection, which in this case is the AppDynamics data source base URL of your account
   - **Orchestration**: Select between 'Local' and 'Remote' radio buttons. If you pick 'Remote', then after you add the integration, you are provided with a bootstrap command to finish the orchestration. Run the bootstrap script on the remote server where you want to deploy the connector remotely.
7. Click **Next**.
8. Enter the following *Collect metric data**: Define how you want to collect metric data.

   - **Enable data collection**: Set the toggle button to on.
   - **Fill out Code field**: Enter Application name and corresponding Metric paths as JSON, using the following format:

     ```
     {"Application Name":"[metric path1, metric path2, metric path3]"
     }
     ```

     Example

     ```
     {
     "helloworld":"[Overall Application Performance|Calls per Minute,Overall Application
     Performance|Average Response Time (ms),Overall Application
     Performance|mytier|Average Response Time (ms)]",
     "products-ecommerce":"[Overall Application Performance|*,Business Transaction
     Performance|Business Transactions|DwightVM|*|*]",
     "Database Monitoring":"[Databases|admin2|KPI|*,Databases|mysql-admin| Server
     ```

```
Statistic|*]"
}
```

To get the metric path, go to the AppDynamics Console, select the application > Metrics Browsers > choose the metrics that you want to monitor > right-click and select 'Copy Full Path'.

- **Client API path**: Enter the API path to get the oauth token, for example `/controller/api/oauth/access_token`.
- **Metrics API path**: Enter the API path to get all applications, for example `/controller/rest/applications`.
- **Historical or Live**: Click either radio button to decide how to collect the data. Historical data start and end date and time must be entered in the following format:
  - **Enter Start/End time**: Enter the start and end times in the format *hh:mm* (hours and minutes).
  - **Set LocalTime**: Pick the time zone required from the dropdown list.
  - **Enter Start/End date**: Enter the Start/End date in the format *MM/DD/YYYY*. **Note**: For historical metric data collection, the recommended time period is two weeks. You can collect up to 31 days of historical data so your start date should not exceed 31 days from the present.

  If you click the 'Live' data radio button the fields to complete are:
  - **Connection interval in minutes for live metrics (1-60)**: currently, the permissible value is 1 min.
  - **Concurrency limit (1-3)**: This is the number of concurrent calls allowed.

9. Click **Done**.

For more information about adding an AppDynamics data retention period, see Adding an AppDynamics data retention period.

For more information about performance considerations for metric data collection, see Performance considerations for metric data collection.

You created an AppDynamics connection in your instance. After you create your connection, you must enable the data collection to connect your connection with the AI of IBM Cloud Pak for AIOps. For more information about enabling your connection, see Enabling AppDynamics connections.

Note: An error that might show if no data is set for particular metrics or applications can resemble the following message.

```
2022-06-22 04:59:07,578 INFO [Default Executor-thread-3]
com.ibm.watson.aiops.connectors.service.AppdynamicsWebClientService : Bad Request
[err] javax.ws.rs.BadRequestException: BadRequest
[err] java.lang.NullPointerException
```

In order to avoid this error, ensure the application name and metrics path configured are valid. Also, ensure that the Client Secret created previously has the following roles: Account Owner and Administrator. If not, the API calls might not return valid application IDs.

To create more connections (such as a ChatOps connection), see Configuring data and tool connections.

For more information about working with the insights provided by your connections, see ChatOps insight management.

Enabling and disabling AppDynamics connections

If you didn't enable your data collection during creation, you can enable your connection afterward. You can also disable a previously enabled connection the same way. To enable or disable a created connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Manage connections* tab of the *Data and tool connections* page, click the **AppDynamics** connection type.
4. Click the connection that you want to enable or disable.
5. Go to the *Collect metric data* section. Set **Enable data collection** to **On** or **Off** to enable or disable data collection. Disabling data collection for a connection does not delete the connection.

You enabled or disabled your connection. For more information about deleting a connection, see Deleting AppDynamics connections.

Editing AppDynamics connections

After you create your connection, your can edit the connection. For example, if you specified *Historical mode* but now want your connection to pull in live data for continuous monitoring, you can edit it. To edit a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **AppDynamics** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the AppDynamics connections page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.
5. Edit your connection as required. Click **Save** when you are done editing.

Your connection is now edited. If your application was not previously enabled or disabled, you can enable or disable the connection directly from the interface. For more information about enabling and disabling your connection, see Enabling and disabling AppDynamics connections. For more information about deleting a connection, see Deleting AppDynamics connections.

Deleting AppDynamics connections

If you no longer need your AppDynamics connection and want to not only disable it, but delete it entirely, you can delete the connection from the console.

To delete a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **AppDynamics** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the AppDynamics connections page, click the options menu (three vertical dots) for the connection that you want to delete and click **Delete**.
5. Enter the name of the connection to confirm that you want to delete your connection. Then, click **Delete**.

Your connection is deleted.

Adding an AppDynamics data retention period

The default data retention period for AppDynamics is 4 hours. Metric anomaly detection uses up to two weeks of data to build models and without historical data to collect, it is several days to a week before anomalies start to be detected and raised without having historical data to quickly train up the algorithms.

Consider whether it is possible to increase the AppDynamics data retention period in advance so that the analytics can train up more quickly. To increase the data retention period in AppDynamics, consult the relevant AppDynamics retention documentation.

Creating AWS CloudWatch connections

An AWS CloudWatch connection collects live and historical metrics data from AWS resources, such as EC2, ECS, or RDS, and sends it to gRPC server as cloud events at specified intervals.

To activate the connection, you must have an endpoint, an access key, and a secret key ready. You can select the metrics that you want to monitor by choosing the associated namespaces in the IBM Cloud Pak Automation console. The Metric anomaly detection AI algorithm analyzes the AWS CloudWatch data.

For more information about working with AWS CloudWatch connections, see the following sections:

- Creating AWS CloudWatch connections
- Enabling and disabling AWS CloudWatch connections
- Editing AWS CloudWatch connections
- Deleting AWS CloudWatch connections

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating AWS CloudWatch connections

To create an AWS CloudWatch connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.
4. From the list of available connections, find and click the **AWS CloudWatch** tile.

   Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Connect**.
6. On the *Add connection* page, define the general connection details:

   - **Name**: Type in the display name of your connection.
   - **Description**: Provide an optional description for the connection.
   - **Endpoint**: Enter an endpoint for the connection.
   - **Access Key**: Enter an Access Key for the connection.
   - **Secret Key**: Enter a Secret Key for the connection. For getting access and secret keys, contact admin.
   - **AWS CloudWatch region**: Select a region from the available regions in AWS, for example, us-east-1.
   - **Deployment options**: Select one of the following options: **Local** or **Remote**. The **Local** option deploys the connection in the same cluster where IBM Cloud Pak for AIOps is installed. Alternatively, the **Remote** option deploys it in your chosen location. **Note**: If you select the **remote** option, you are provided with a bootstrap command after adding the connection to finish the deployment.
7. Click **Test connection**. A message stating whether the test connection succeeded or failed is displayed.
8. Click **Next**.
9. Enter the following *Collect metric data* information. It defines how you want to collect metric data.

   - **Enable data collection**: Set the toggle button to on. **Note**: If you disable data flow, a new connection is created, but metrics are not collected.
   - **Namespace**: Select one or more namespaces from the list to see its associated metrics in the filter table. By default, all metrics are selected in the table. You can search and select or remove metrics from it.
   - **Aggregation interval (minutes)**: Enter a value in minutes. This value is the frequency at which data is collected. The default value is 5 minutes.
   - Select live or historical data collection **Note**: If you choose to collect only live data, you cannot collect historical data after the live data collection starts. However, if you collect historical data, you can continue live data collection afterwards.
     - Live data collection: To collect live data, complete the following steps:
       - **Only collect live data once the connection is established**: Select the checkbox.
       - **Scheduler run time (minutes)**: Enter a value in minutes. The value represents the frequency at which live scheduler runs. For example, if you set the value to 10 minutes, the live scheduler runs every 10 minutes. The default value is 5 minutes.
     - Historical data collection: To collect historical data, complete the following steps:

- **Start date and time**: Select the start date from the calendar and enter the time in the 'hh:mm' format (hours and minutes). **Note**: The start date must not exceed 31 days from the present as the maximum time period for historical data collection is 31 days. The recommended time period is two weeks.
- **Time zone**: Select your time zone from the dropdown list.
- **End date and time**: Select a date from the calendar. Also, enter the time in the hh:mm format (hours and minutes). If you want to the end date, click **Remove end date**. **Note**: If you do not set an end date, IBM Cloud Pak for AIOps collects historical data from the specified start date and continues with live data collection. You cannot collect historical data after the live data collection starts.
- **Scheduler run time (minutes)**: Enter a value in minutes. The value represents the frequency at which live scheduler runs. The default value is 5 minutes. **Note**: Scheduler run time is only applicable to live mode. You must configure the scheduler run time only if you do not add an end date as the connector will continue to run in live mode after historical mode.

10. Click **Done**.

For more information about performance considerations for metric data collection, see Performance considerations for metric data collection.

Enabling and disabling AWS CloudWatch connections

If you didn't enable your data collection during creation, you can enable your connection afterward. You can also disable a previously enabled connection the same way. To enable or disable a created connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Manage connections* tab of the *Data and tool connections* page, click the **AWS CloudWatch** connection type.
4. Click the connection that you want to enable or disable.
5. Go to the *Collect metric data* section. Set **Enable data collection** to **On** or **Off** to enable or disable data collection. Disabling data collection for a connection does not delete the connection.

You enabled or disabled your connection. For more information about deleting a connection, see Deleting AWS CloudWatch connections.

Editing AWS CloudWatch connections

After you create your connection, you can edit it. To edit a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **AWS CloudWatch** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the AWS CloudWatch connections page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.
5. Edit your connection as required. Click **Save** when you are done editing.

Deleting AWS CloudWatch connections

If you no longer need your AWS CloudWatch connection and want to not only disable it, but delete it entirely, you can delete the connection from the console.

To delete a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.

2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **AWS CloudWatch** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the AWS CloudWatch connections page, click the options menu (three vertical dots) for the connection that you want to delete and click **Delete**.
5. Enter the name of the connection to confirm that you want to delete your connection. Then, click **Delete**.

Your connection is deleted.

Creating custom connections

A custom connection ingests user-defined log data, which is used to establish a baseline of normal behavior and then identify anomalies. These anomalies can be correlated with other alerts and events and published to your ChatOps interface, which help you determine the cause and resolution of a problem.

Due to each user-defined data source having different formats and sizes, not all log sources can be ingested directly with this custom connection. For more information, see Data source requirements.

Important: Custom connections cannot use historical data for initial AI training.

Note: As an alternative to using the **Custom** connector that is provided by default, you can create your own connector to interact with IBM Cloud Pak for AIOps. For instance you can create a custom connector for sending events, metrics, and topology data to IBM Cloud Pak for AIOps from an external source when a default connector is not available. For more information see Creating custom connections using the Connector SDK

For more information about working with custom connections, see the following sections:

- Data source requirements
- Creating custom connections
  - Customizing `GET` and `POST` connections
  - Custom mappings
- Enabling custom connections
- Editing custom connections
- Deleting custom connections

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Data source requirements

Before creating the connection, you should be aware of the following information.

- **Format of source data**: The data that is ingested with the log connector must be in newline-delimited JSON format. Each individual log entry must be on a separate line and each log entry must be less than 1 MB, otherwise the custom connection is not able to ingest the data. The following code snippet shows an example of the newline-delimited JSON format:

  ```
  {"message": "exampleMessage", "time": 500}

  {"message": "exampleMessage", "time": 500}
  ```

  Custom connections support the ingestion of flat JSON source data but do not support nested key-value pairs. The following examples illustrate what is supported and what is not.

  Supported JSON source data: the following code snippet shows an example of flat JSON source data.

  ```
  {
     "message": "exampleMessage",
     "time": "..."
  }
  ```

Unsupported JSON source data: the following code snippet shows an example of nested key-value pairs.

```
{
   "log":["message": "exampleMessage","timezone": "exampleTimezone"],
   "time": "..."
}
```

Important: If your data does not conform with the preceding format, the custom connection is not be able to ingest the data. Review the requirements before creating your custom connection.

- **Load**: To prevent this connection placing an inordinate load on your data source and potentially impacting your logging operations, this connection only connects to one API with a default data frequency of 60 seconds. This is controlled by using the **Sampling rate** setting in the Procedure section.
- **Access**: Custom data sources are cloud-based REST APIs. Access is configured by using the authentication methods that are specified in the **Authentication type** setting in the Procedure section.
- **Data volume**: Data volume depends on the application, and is not a set value. Therefore, it does not appear in the settings.

Note: If you want to collect log data from multiple systems and then route that information through a forwarding agent, such as an Apache Kafka topic (for example, Sysdig or FluentBit), you must use the Kafka connection type. For more information about Kafka connections, see Creating Kafka connections.

Creating custom connections

The custom connection type collects newline-delimited JSON from a single specific data source.

Procedure

To create a custom connection from a specific source, step through the following sections:

- Adding a connection
- Specifying connection parameters
- Specifying field mapping
- Specifying how log data is collected for AI training

Adding a connection

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.
4. From the list of available connections, find and click the **Custom** tile.

   Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Connect**.

Specifying connection parameters

1. On the *Add connection page* enter the following connection information:
   - **Name**: The display name of your connection.
   - **Description**: An optional description for the connection.
   - **Extract data from an array of log lines embedded in JSON**: When enabled, logs are extracted from an array that is embedded within a JSON object that is returned by the REST service. This option works only for an array of flat JSON objects that is embedded within a JSON object.

- **Embedded array path**: Type the path to the array from which you want to extract log data. The following examples, show how to configure the array path:

  ```
  {"results": [{"message": "myMessage"}]} would be results[*]
  {"results": {"events": [{"message": "myMessage"}]}} would be results.events[*]
  {"results": [{"events": [{"message": "myMessage"}]}]} would be results[0].events[*]
  ```

- **REST method**: The method by which your data gets transferred, either **`GET`** or **`POST`**.
- **Authentication type**: Select one of the following values:
  - **User ID/password**: The user ID and password that is required for accessing the REST service. You must enter both in the connection configuration. The HTTPS authentication header is sent to the Service URL in the format 'Authorization: Basic MyCredential'. 'MyCredential' is base64 encoded and in the format 'username:password'.
  - **API key**: The REST service is authenticated with an API key. The HTTPS authentication header is sent to the Service URL in the format 'Authorization: MyAPIKey'.
  - **Token**: The REST service is authenticated with a temporary token. The HTTPS authentication header is sent to the Service URL in the format 'Authorization: Bearer MyToken'.
  - **Custom**: Your custom REST service has specific authentication requirements that are not covered by the other authentication methods. You must enter both a Header and Value to configure your authentication. Note, both the Header and Value are plain text properties that are case-sensitive and does not remove spaces. The HTTPS authentication header is sent to the Service URL in the format 'Authorization: MyHeader MyValue'.
  - **None**: Your custom REST service requires no authentication. No Authorization header is sent to the target system.
  - **Query Key**: The REST service is authenticated with a query key. The HTTPS authentication header is sent to the Service URL in the format **`X-Query-Key: MyQueryKey`**.
- **Certificate (optional)**: Certificate used to verify the SSL/TLS connection to the REST service.
- **UI URL**: The hostname or IP address of the UI of your custom source.
- **Service URL**: The URL for the service that you want to integrate. For more information about customizing your *Service URL*, see Customizing GET and POST connections
- **Base parallelism**:It is recommended to use a higher value than 1 so that you can process data in parallel. In a small environment the available flinks slots is 16 and for a large environment, the maximum available slots is 32.
- **Sampling rate**: The rate at which data is pulled from live source (in seconds). The default value is **`60`**.
- **JSON processing option**: Select a JSON processing option.
  - **None**: The default option. The JSON is not processed or modified.
  - **Flatten**: This option flattens the JSON object by removing the opening and closing braces.
  - **Filter**: This option extracts the JSON object and replaces it with an empty string.
  - For more information about the options, see Managing embedded JSON.

Note: To improve data throughput, you can increase the base parallelism value incrementally. For more information about maximum base parallelism for starter and production deployments, see Improving data streaming performance for log anomaly detection.

2. Click **Next** to move to the next page.

Specifying field mapping

1. On the *Field mapping* page you can improve search performance by mapping the fields from your implementation fields to standard fields within IBM Cloud Pak for AIOps.

- For more information about how field mappings are defined, see Mapping data from incoming sources.
- For more information about using mappings to clean your data for use in IBM Cloud Pak for AIOps, see Cleaning mapped data using regular expressions.

The following code snippet displays an example of field mapping that uses the supported format. When you are coding your mapping, use this example to help you.

```
{
  "codec": "custom",
  "message_field": "message",
  "log_entity_types": "kubernetes.container_name",
  "instance_id_field": "_app",
  "timestamp_settings":{
     "timestamp_field":"_ts", <MANDATORY>
     "multiplier": 1000, <OPTIONAL>
     "pattern": "yyyy-MM-dd hh:mm:ss z" <OPTIONAL, SPECIFIC TO RAW DATA OUTPUT>
  }
}
```

If you are configuring a connection to WebSphere, then you must use the following mapping. In this mapping, all fields must be specified exactly as shown, except for the **log_entity_types** parameter. This can be empty, as shown in the following example.

```
{
"codec": "custom",
"rolling_time": 10,
"instance_id_field": "ibm_serverName",
"log_entity_types": "",
"message_field": "message",
"timestamp_settings": {
   "timestamp_field": "ibm_datetime",
   "pattern": "yyyy-MM-dd'T'HH:mm:ss.SSSZZZZZ"
                      }
}
```

For more information about specific custom mappings, see Custom mappings.

Alternatively, if you want to include specific fields from your WebSphere record within the log alert **resource** field to be generated by the log anomaly detection algorithms, then specify those WebSphere fields in the **log_entity_types** parameter, as shown in the example below.

```
{
"codec": "custom",
"rolling_time": 10,
"instance_id_field": "ibm_serverName",
"log_entity_types": "host,ibm_userDir,ibm_serverName,module",
"message_field": "message",
"timestamp_settings": {
   "timestamp_field": "ibm_datetime",
   "pattern": "yyyy-MM-dd'T'HH:mm:ss.SSSZZZZZ"
                      }
}
```

For more information about the log alerts generated by the log anomaly detection algorithms, see the following topics:

- Natural language examples
- Statistical baseline examples

You can test your connection by clicking **Test connection**.

2. Click **Next** to move to the next page.

Specifying how log data is collected for AI training

1. On the *AI training and log data* page, select how you want to manage collecting data for use in AI training and anomaly detection. Click the *Data collection* toggle to turn on data collection then select how you want to collect data:

   - **Live data for continuous AI training and anomaly detection**: A continuous collection of data from your connection is used to both train AI models and analyze your data for anomalous behavior.

Note: After an initial installation, there is no data at all in the system. If you select this option, then the two different log anomaly detection algorithms behave in the following ways:

- Natural language log anomaly detection does not initially detect anomalies as no model has been trained. If **Live data for continuous AI training and anomaly detection** is set to on, the system gathers training data live and after a few days there is enough data to train a model. When this model is deployed, then it detects anomalies as normal.
- Statistical baseline log anomaly detection does not detect anomalies for the first 30 minutes of data collection. This is because it does not have a baseline yet. After 30 minutes of live data collection the baseline is automatically created. After that it detects anomalies on an ongoing basis, while continuing to gather data and improve its model every 30 minutes.

- **Live data for initial AI training**: A single set of training data used to define your AI model. Data collection takes place over a specified time period that starts when you create your connection.

  Note: Selecting this option causes the system to continue to collect data while the option is enabled; however, the data is collected for training only, and not for log anomaly detection. For more information about AI model training, including minimum and ideal data quantities, see Configuring AI training.

2. **Important**: Keep in mind the following considerations when you select your data collection type:
   - Anomaly detection for your connection occurs if you select *Live data for continuous AI training and anomaly detection*.
   - Custom connections cannot use historical data for AI training.
   - Different types of AI models have different requirements to properly train a model. Make sure that your settings satisfy minimum data requirements. For more information about how much data you need to train different AI models, see Configuring AI training.
3. Click **Done**.

You have created a custom connection in your instance. After you create your connection, you must enable the data collection to connect your connection with the AI of IBM Cloud Pak for AIOps. For more information about enabling your connection, see Enabling custom connections.

To create more connections (such as a ChatOps connection), see Configuring data and tool connections.

For more information about working with the insights provided by your connections, see ChatOps insight management.

Customizing `GET` and `POST` connections

`GET` connections

If you customize the *Service URL* of your connection, you must include the start and end time in the URL with properly formatted timestamps (`SimpleDateFormat`). For epoch timestamps, milliseconds and seconds are supported. The following timestamp identifiers are supported in the **Service URL** field:

- <start-epoch-ms>, <end-epoch-ms>. For example, `http://exampleurl.com/v1/export?to=<end-epoch-ms>&from=<start-epoch-ms>`.
- <start-epoch-s>, <end-epoch-s>. For example, `http://exampleurl.com/v1/export?to=<end-epoch-s>&from=<start-epoch-s>`.
- <start-`SimpleDateFormat`>, <end-`SimpleDateFormat`>. For example, `http://exampleurl.com/v1/export?to=<end-yyyy-MM-dd'T'HH:mm:ss.SSSXXX>&from=<start-yyyy-MM-dd'T'HH:mm:ss.SSSXXX>`.

`POST` connections

For `POST` connections, you do not need to customize the *Service URL* value. For these connections, the payload is in the body of the JSON and you can customize the format, but must include the start and end times with properly formatted timestamps (`SimpleDateFormat`). For epoch timestamps, milliseconds and seconds are supported. The following timestamp identifiers are supported in the **Service URL** field:

- <start-epoch-ms>, <end-epoch-ms>. For example, `{"queryString":"exampleQuery","start":"<start-epoch-ms>", "end":"<end-epoch-ms>","isLive":false}`

- <start-epoch-s>, <end-epoch-s>. For example, `{"queryString":"exampleQuery","start":"<start-epoch-s>", "end":"<end-epoch-s>","isLive":false}`
- <start-**SimpleDateFormat**>, <end-**SimpleDateFormat**>. For example, `{"query": {"range": {"@timestamp": {"gte": "<start-yyyy-MM-dd'T'HH:mm:ss.SSSXXX>","lt": "<end-yyyy-MM-dd'T'HH:mm:ss.SSSXXX>"}}},"size": 10000}`

Custom mappings

Custom mappings allow more flexibility regarding timestamp settings. You must provide the mapping information about the timestamp field and how to parse the timestamp. Custom mappings support unnested JSON inputs.

The basic building blocks are similar to other connections. The **rolling_time**, **instance_id_field**, **log_entity_types**, and **message_field** mappings are defined in the same way. For custom connections, an extra **timestamp_settings** mapping needs to be specified. It contains the required **timestamp_field** to tell what field contains the timestamp, and two optional fields **pattern** and **multiplier**. The **pattern** field must be used when the timestamps must be parsed for a **String** representation, and the m**ultiplier** field must be used when the timestamp is represented as the **Double**type.

When timestamps are formatted correctly, as 13-digit UNIX® timestamps, you do not need to specify the **pattern** or **multiplier** fields. You need to define the **timestamp_field** field as in the following example:

```
{
    "rolling_time": 10
    "instance_id_field": "_app",
    "log_entity_types": "kubernetes.container_name",
    "message_field": "message",
    "timestamp_settings":{
      "timestamp_field":"_ts"
    }
{
```

Custom mapping with String type timestamps

When timestamps are represented as Strings, they must be parsed and converted to 13-digit UNIX timestamps. In this case, the **pattern** field must contain the parsing pattern. For example, if the timestamp is **2020-12-31 23:59:59 PST**, the connection mapping can be defined as follows:

```
{
    "rolling_time": 10,
    "instance_id_field": "_account",
    "log_entity_types": "kubernetes.container_name",
    "message_field": "_line",
    "timestamp_settings":{
      "timestamp_field":"_ts",
      "pattern": "yyyy-MM-dd hh:mm:ss z"
    }
}
```

Custom mapping with Double type timestamps

It might occur that timestamps are not represented in milliseconds. In this case, you must multiply or divide the input timestamps to obtain the milliseconds. The following example illustrates how you can use the **multiplier** field to obtain milliseconds.

Consider the following source timestamp without milliseconds:

```
{
   "_ts": 1569873609.539,
   "_account": "...",
   "_cluster": "...",
   "_host": "...",
   "_logtype": "...",
   "_line": "...",
   "_id": "...",
   "__key": "...",
}
```

You can set the **multiplier** field in the mapping to show the timestamps in milliseconds:

```
{
    "rolling_time": 10,
    "instance_id_field": "_account",
    "log_entity_types": "kubernetes.container_name",
    "message_field": "_line",
    "timestamp_settings":{
      "timestamp_field":"_ts",
      "multiplier": 1000
    }
}
```

Custom mapping with static instance ID support

You can input a static value for the **instance_id_field** in the IBM Cloud Pak for AIOps pipeline. See the following example:

```
{
   "rolling_time":10,
   "instance_id_field":"%aStaticInstanceId%",
   "log_entity_types":"kubernetes.container_name",
   "message_field":"_line",
   "timestamp_settings":{
      "timestamp_field":"_ts",
      "pattern":"yyyy-MM-dd hh:mm:ss z"
   }
}
```

The custom mapping places the value that is contained between **%...%** in the **instance_id_field** field throughout the IBM Cloud Pak for AIOps data collection. The following example illustrates the resulting data from the preceding custom mapping:

```
{
   "message":"exampleLogMessage",
   "instance_id":"aStaticInstanceId",
   "entities":{
      "pod":"examplePod",
      "_cluster":"exampleCluster",
      "container":"exampleContainer"
   },
   ...,
   "timestamp":1605195377154
}
```

Supported custom-mapping fields

Custom mapping fields

Field Name	Description	Default Value
rolling_time	The recurring time interval to capture data from in seconds.	10
instance_id_field	The field name in the incoming data that contains the application name.	kubernetes.namespace_name, kubernetes.host, kubernetes.container_name
log_entity_types	The field name in the incoming data that contains the entity to extract and analyze.	kubernetes.container_name
message_field	The field name in the incoming data that contains the log message.	**@rawstring**
standard_entity_types	Standardized fields. If declared, the performance for event grouping can be improved.	**None**
standard_entity_types.pod_name	Standardized field. If declared, the performance for event grouping can be improved.	**pod**
standard_entity_types.node_name	Standardized field. If declared, the performance for event grouping can be improved.	**node**
timestamp_settings	Customizable settings on timestamps from source's raw log output.	**None**
timestamp_settings.timestamp_field	The field name in the incoming data that contains the data's timestamp.	**_ts**

Field Name	Description	Default Value
timestamp_settings.pattern (Optional)	Must be used when the timestamps must be parsed for a string representation. Custom connections support `SimpleDateFormat` for timestamps in incoming data.	`yyyy-MM-dd hh:mm:ss z`
timestamp_settings.multiplier (Optional)	Used when epoch timestamps are represented as a double.	`None`

Enabling and disabling custom connections

If you didn't enable your data collection during creation, you can enable your connection afterward. You can also disable a previously enabled connection the same way. If you selected *Live data for initial AI training* when you created your connection, you must disable the connection before AI model training. To enable or disable a created connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Manage connections* tab of the *Data and tool connections* page, click the **Custom** connection type.
4. Click the connection that you want to enable or disable.
5. Go to the *AI training and log data* section. Set **Data collection** to **On** or **Off** to enable or disable data collection. Disabling data collection for a connection does not delete the connection.

Your connection is now enabled or disabled. For more information about deleting a connection, see Deleting custom connections.

Editing custom connections

After you create your connection, you can edit the connection. To edit a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Custom** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the Custom connections page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.
5. Edit your connection as required. Click **Save** when you are done editing.

Your connection is now edited. If you have not enabled or disabled your application, you can enable or disable the connection directly from the interface. For more information about enabling and disabling your connection, see Enabling and disabling custom connections. For more information about deleting a connection, see Deleting custom connections.

Deleting custom connections

If you no longer need your custom connection and want to not only disable it, but delete it entirely, you can delete the connection from the console.

Note: You must disable data collection before deleting your connection. For more information about disabling data collection, see Enabling and disabling custom connections.

To delete a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.

3. Click the **Custom** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the Custom connections page, click the options menu (three vertical dots) for the connection that you want to delete and click **Delete**.
5. Enter the name of the connection to confirm that you want to delete your connection. Then, click **Delete**.

Your connection is deleted.

Creating custom connections using the Connector SDK

You can create your own connectors to interact with IBM Cloud Pak for AIOps, such as for sending events, metrics, and topology data to IBM Cloud Pak for AIOps from an external source when a default connector is not available.

The framework for creating connectors is documented in the following repository, which also hosts the Java template for new custom connectors for IBM Cloud Pak for AIOps: IBM/cp4waiops-connectors-java-template. This connector template provides you with a working example for extending IBM Cloud Pak for AIOps. You can use this example as a starting base for developing your own custom connection. Details are included in the repository documentation to help you understand the connector framework architecture and to help you proceed from development to deployment.

The connector Java template uses the Connector SDK, which can be obtained from Maven central. This Connector SDK abstracts away the gRPC logic, provides security, and provides common methods to help you accelerate your custom connector development. Check Maven central for the latest version of the Connector SDK before you create or update your custom connections.

With the Connector SDK and template, you can create your own connector for sending the following types of data to IBM Cloud Pak for AIOps:

- Events
- Metrics
- Topology

Creating Dynatrace connections

Dynatrace connections collect different types of metrics, including Live and Historical data, and send them to the gRPC server as Cloud events at specified intervals. You need to have both an endpoint and an API token ready. The AI Algorithms that analyze Dynatrace data are probable cause, and temporal correlation. Dynatrace SaaS 1.247 with v2 Metrics API is the version that is supported for the Dynatrace Connector.

For more information about working with Dynatrace connections, see the following sections:

- Creating Dynatrace connections
- Enabling Dynatrace connections
- Editing Dynatrace connections
- Deleting Dynatrace connections

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating Dynatrace connections

To create a Dynatrace connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.

4. From the list of available connections, find and click the **Dynatrace** tile.

   Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section.

5. On the side-panel, review the instructions and when ready to continue, click **Connect**.

6. On the *Add connection* page, define the general connection details:

   - **Name**: The display name of your connection. You cannot use spaces or special characters in the display name. Use only alphanumeric characters.
   - **Description**: An optional description for the connection.
   - **Endpoint**: Dynatrace data source base url. For example, `https://<your-environment-id>.live.dynatrace.com`.
   - **Access Token**: Enter an API key for the connection.
   - **Orchestration**: Select between 'Local' and 'Remote' radio buttons. If you pick 'Remote', then after you add the integration, you are provided with a bootstrap command to finish the orchestration.

7. Click **Next**.

8. Enter the following *Collect metric data* (Optional): Define how you want to collect metric data.

   - **Enabled data collection**: Set the 'Enabled data collection' toggle button to 'on'.

     Note: A monitored Metrics note appears telling you that you can select built-in metrics to monitor. These metrics are categorized by the technologies that they are associated with. You can also add custom-defined metrics to monitor.

     You are asked to select a technology to see the list of associated built-in metrics. These technologies include 'Billing', 'Cloud', 'Containers', 'Dashboard' and so on. Depending on which you select the built-in metrics selection varies.

   - **Metric filter**: If you click the 'Built in Metric Name' checkbox all metrics are checked for the particular Technology you selected in the dropdown menu, and you see the number of possible metrics listed.

   - **Additional Metrics**: Enter any additional metrics in the free text box. These are custom-defined metrics such as strings separated by commas. For example, metric1, metric2.

   - **Metrics API path**: Enter the API path to get all applications, for example '/api/v2/metrics/query'

   - **Historical or Live**: Click either radio button to decide how to collect the data. Historical data start and end date and time must be entered in the following format:

     - **Enter Start/End time**: Enter the start and end time in the format "hh:mm" (hours and minutes).
     - **Localtime**: Pick the time zone from the dropdown list.
     - **Enter Start/End date**: Enter the start and end date in the format "MM/DD/YYYY".

     Note: For historical metric data collection, the recommended time period is two weeks, though you can collect up to 31 days of historical data so your start date should not exceed 31 days from the present.

   If you clicked in the 'Live' Data radio button the fields that you are required to complete are:

   - **Connection interval in minutes for live metrics (1-60)**: The default value here in 1 min.
   - **Concurrency limit (1-3)**: This is the number of concurrent calls allowed.

9. Click **Done**.

Note: The API limit can cause a delay in receiving metrics from Dynatrace. Each Dynatrace token is limited to 50 API requests per minute. Multiple API requests might be needed for each metric data collection interval based on the amount of data that Dynatrace returns. If the required requests to the API exceed the limit, the integration throttles the requests until more requests are available in the next minute. Review the performance considerations for metric integrations and the filtering options available to optimize this behavior. For more information, see Performance considerations for metric data collection.

You created a Dynatrace connection in your instance. After you create your connection, you must enable the data collection to connect your connection with the AI of IBM Cloud Pak for AIOps. For more information about enabling your connection, see Enabling Dynatrace connections.

To create more connections (such as a ChatOps connection), see Configuring data and tool connections.

For more information about working with the insights provided by your connections, see ChatOps insight management.

Enabling and disabling Dynatrace connections

If you didn't enable your data collection during creation, you can enable your connection afterward. You can also disable a previously enabled connection the same way. To enable or disable a created connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Manage connections* tab of the *Data and tool connections* page, click the **Dynatrace** connection type.
4. Click the connection that you want to enable or disable.
5. Go to the *Collect metric data* section. Set **Enable data collection** to **On** or **Off** to enable or disable data collection. Disabling data collection for a connection does not delete the connection.

You enabled or disabled your connection. For more information about deleting a connection, see Deleting Dynatrace connections.

Editing Dynatrace connections

After you create your connection, your can edit the connection. For example, if you specified *Historical data for initial AI training* but now want your connection to pull in live data for continuous monitoring, you can edit it. To edit a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Dynatrace** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the Dynatrace connections page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.
5. Edit your connection as required. Click **Save** when you are done editing.

Your connection is now edited. If your application was not previously enabled or disabled, you can enable or disable the connection directly from the interface. For more information about enabling and disabling your connection, see Enabling and disabling Dynatrace connections. For more information about deleting a connection, see Deleting Dynatrace connections.

Deleting Dynatrace connections

If you no longer need your Dynatrace connection and want to not only disable it, but delete it entirely, you can delete the connection from the console.

To delete a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Dynatrace** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the Dynatrace connections page, click the options menu (three vertical dots) for the connection that you want to delete and click **Delete**.
5. Enter the name of the connection to confirm that you want to delete your connection. Then, click **Delete**.

Your connection is deleted.

Creating ELK connections

Elasticsearch, Logstash, and Kibana (ELK) connections provide log data for anomaly detection in IBM Cloud Pak® for AIOps. An ELK connection provides log data, which is used to establish a baseline of normal behavior and then identify anomalies. These anomalies can be correlated with other alerts and events, and published to your ChatOps interface to help you determine the cause and resolution of a problem.

For more information about working with ELK connections, see the following sections:

- Creating ELK connections
- Enabling ELK connections
- Editing ELK connections
- Deleting ELK connections

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

About this task

Before creating the connection, you should be aware of the following information.

- **Load**: To prevent this connection placing an inordinate load on your data source and potentially impacting your logging operations, this connection only connects to one API with a default data frequency of 60 seconds. This is controlled by using the **Sampling rate** setting in the Procedure section.
- **Access**: Custom data sources are cloud-based REST APIs. Access is configured by using the authentication methods that are specified in the **Authentication type** setting in the Procedure section.
- **Data volume**: Data volume depends on the application, and is not a set value. Therefore, it does not appear in the settings.

Known issue with later versions of ELK

IBM Cloud Pak for AIOps only supports ELK up to version 7.12.0. For later versions of ELK, the **`event_link`** provided by AIOps does not open the correct logs in Kibana.

The reason for this incorrect link is that later versions of ELK use the **`uuid`** as the default ID for the index pattern. As a workaround, you must set the Custom index pattern ID to same value as the index pattern name.

To do so, use the following steps:

1. Log in to the ELK instance.
2. From the hamburger menu, navigate to **Stack Management**.
3. Click on **Create index pattern**.
4. Complete the required details.
5. Click on the **Show advanced settings button**.
6. Set the **Custom index pattern ID** to the same value that you specified for the **Index pattern name**. This will override the default value.

Procedure

To create an ELK connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.
4. From the list of available connections, find and click the **ELK** tile.

Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section.

5. On the side-panel, review the instructions and when ready to continue, click **Connect**.
6. On the *Add connection* page, define the general connection details:
    - **Name**: The display name of your connection.
    - **Description**: An optional description for the connection.
    - **ELK service URL**: ELK service URL: The Elasticsearch host and public API port. The URL must have the target index that IBM Cloud Pak for AIOps uses to search for the data from your applications. If you want to use all data, specify `*` as your index. For example, your ELK service URL can be `https://myURL.com:8080/*`.
    - **Kibana URL**: Enter a URL for the service instance.
    - **Authentication type**: Select one of the following values:
        - **User ID/password**: The Elasticsearch instance has a user ID and password as authentication. You must enter both in the connection configuration.
        - **API key**: The Elasticsearch instance is authenticated with an API key.
        - **Token**: The Elasticsearch instance is authenticated with a temporary token.
        - **None**: The Elasticsearch instance has no authentication.

      Note: If you selected API key as Authentication type value, then you need to follow the steps on ELK connector steps to use ApiKey authentication method.
    - **User ID**: Enter a user ID for the connection.
    - **Password**: Enter a password for the connection.
    - **Certificate (optional)**: Certificate used to verify the SSL/TLS connection to the REST service.
    - **Filters (optional)**: A custom Boolean Query to filter the Elasticsearch request for your specific application, terms, keywords, or other filters.
    - **Time zone (optional)**: The time zone in which your data is situated. The default time is converted from the system time relative to UTC time. The default value is UTC.
    - **Kibana port**: The port of the Kibana instance that is on the same host as the Elasticsearch instance. If the Kibana instance is not exposed on the default `5601` port, change this value to the port where the Kibana instance is shown.
    - **Base parallelism**: Select a value to specify the number of Flink jobs that can run in parallel. These jobs run to process and normalize the collected data. The default value is 1. However, it is recommended to use a higher value than 1 so that you can process data in parallel. This value cannot exceed the total available free Flink slots. In a small environment, the available flinks slots are 16, while in a large environment, the maximum available slots are 32. If you are collecting historical data with this connection, you can set this value to be equal to the **source** parallelism.
    - **Sampling rate**: The rate at which data is pulled from live source (in seconds). The default value is `60`.
    - **JSON processing option**: Select a JSON processing option.
        - **None**: The default option. The JSON is not processed or modified.
        - **Flatten**: This option flattens the JSON object by removing the opening and closing braces.
        - **Filter**: This option extracts the JSON object and replaces it with an empty string.
        - For more information about the options, see Managing embedded JSON.

    Note: To improve data throughput, you can increase the base parallelism value incrementally. For more information about maximum base parallelism for starter and production deployments, see Improving data streaming performance for log anomaly detection.

    Note: If using the Filter option, you should not use a timestamp in the filter query. It gives a parsing error in the backend.

```
    "range": {
      "@timestamp": {
        "gte": "now-2m",
        "lt" : "now"
       }
```

Other than that, ELK filter can use any clauses so long as the fields and the values that are specified in the filter are relevant to the target endpoint data set.

Filter debugging tip: You can first test the filter by using the curl command to ELK endpoint. When using the curl command, the timestamp range is optional in the elk filter, and depends on whether you want to narrow down the search over a particular period of time. If the query gets zero records with 200 HTTP responses (for example no filter error), it is an indication that the query was not able to hit the particular set of data scoped by the filter. Users can tune filter to get the needed results.

7. You can test your connection by clicking **Test connection**.
8. Click **Next**.
9. Enter *Field Mapping* information (Optional):

   You can improve search performance by mapping the fields from your implementation fields to IBM Cloud Pak for AIOps's standard fields. For more information about how field mappings are defined, see Mapping data from incoming sources. For more information about using mappings to clean your data for use in IBM Cloud Pak for AIOps, see Cleaning mapped data that use regular expressions. Consider the supported data schema when you create your field mapping:

```
{
  "codec": "elk",
  "rolling_time": 10,
  "instance_id_field": "application_name",
  "log_entity_types": "kubernetes.pod_name",
  "message_field": "message",
  "timestamp_field": "@timestamp",
  "resource_id": "kubernetes.pod_name"
}
```

10. Click **Next**.
11. Enter AI training and log data (Optional):

    Select how you want to manage collecting data for use in AI training and anomaly detection. Click the **Data collection** toggle to turn on data collection, then select how you want to collect data:

    - **Live data for continuous AI training and anomaly detection**: A continuous collection of data from your connection is used to both train AI models and analyze your data for anomalous behavior.

      Note: After an initial installation, there is no data at all in the system. If you select this option, then the two different log anomaly detection algorithms behave in the following ways:

      - Natural language log anomaly detection does not initially detect anomalies as no model has been trained. You can retrieve historical data (select **Historical data for initial AI training**) to speed up the retrieval of data to train on, or you can leave the **Live data for continuous AI training and anomaly detection** setting on. In the latter case, the system gathers training data live and after a few days there is enough data to train a model. When this model is deployed, then it detects anomalies as normal.

      - Statistical baseline log anomaly detection does not detect anomalies for the first 30 minutes of data collection. This is because it does not have a baseline yet. After 30 minutes of live data collection the baseline is automatically created. After that it detects anomalies on an ongoing basis, while continuing to gather data and improve its model every 30 minutes.

    - **Live data for initial AI training**: A single set of training data used to define your AI model. Data collection takes place over a specified time period that starts when you create your connection.

      Note: Selecting this option causes the system to continue to collect data while the option is enabled; however, the data is collected for training only, and not for log anomaly detection. For more information about AI model training, including minimum and ideal data quantities, see Configuring AI training.

- **Historical data for initial AI training**: A single set of training data used to define your AI model. You need to give Start and End dates, and specify the parallelism of your source data. Historical data is harvested from existing logs in your connection over a specified time period in the past.
    - **Start date**: Select a start date from the calendar and enter the time in hh:mm (hours and minutes) format.

      Note: The start date must not exceed 31 days from the present as the maximum time period for historical data collection is 31 days. The recommended time period is two weeks.
    - **Time zone**: Select your time zone from the dropdown list.
    - **End date and time**: Click **Add end date** and select an end date from the calendar and enter the time in hh:mm format.

      Note: If you do not specify the end date, then live data collection follows the historical data collection. If you do not want to set an end date, click **Remove end date**.
    - **Source parallelism (1-50)**: Select a value to specify the number of requests that can run in parallel to collect data from the source. Generally, you can set the value to equal the number of days of datat that you want to collect. When you are setting this value, consider the number of requests that are allowed by the source in a minute. For example, if only 1-2 requests are allowed, set the value to be low.

Important: Keep in mind the following considerations when you select your data collection type:

- Anomaly detection for your connection occurs if you select *Live data for continuous AI training and anomaly detection*.
- Different types of AI models have different requirements to properly train a model. Make sure that your settings satisfy minimum data requirements. For more information about how much data you need to train different AI models, see Configuring AI training.

12. Click **Done**.

You created an ELK connection in your instance. After you create your connection, you must enable the data collection to connect your connection with the AI of IBM Cloud Pak for AIOps. For more information about enabling your connection, see Enabling ELK connections.

To create more connections (such as a ChatOps connection), see Configuring data and tool connections.

For more information about working with the insights provided by your connections, see ChatOps insight management.

Enabling and disabling ELK connections

If you didn't enable your data collection during creation, you can enable your connection afterward. You can also disable a previously enabled connection the same way. If you selected *Live data for initial AI training* when you created your connection, you must disable the connection before AI model training. To enable or disable a created connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Manage connections* tab of the *Data and tool connections* page, click the **ELK** connection type.
4. Click the connection that you want to enable or disable.
5. Go to the *AI training and log data* section. Set **Data connection** to **On** or **Off** to enable or disable data collection. Disabling data collection for a connection does not delete the connection.

You enabled or disabled your connection. For more information about deleting a connection, see Deleting ELK connections.

Editing ELK connections

After you create your connection, your can edit the connection. For example, if you specified *Historical data for initial AI training* but now want your connection to pull in live data for continuous monitoring, you can edit it. To edit a connection,

complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **ELK** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the ELK connections page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.
5. Edit your connection as required. Click **Save** when you are done editing.

Your connection is now edited. If your application was not previously enabled or disabled, you can enable or disable the connection directly from the interface. For more information about enabling and disabling your connection, see Enabling and disabling ELK connections. For more information about deleting a connection, see Deleting ELK connections.

Deleting ELK connections

If you no longer need your ELK connection and want to not only disable it, but delete it entirely, you can delete the connection from the console.

Note: You must disable data collection before deleting your connection. For more information about disabling data collection, see Enabling and disabling ELK connections.

To delete a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **ELK** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the ELK connections page, click the options menu (three vertical dots) for the connection that you want to delete and click **Delete**.
5. Enter the name of the connection to confirm that you want to delete your connection. Then, click **Delete**.

Your connection is deleted.

Generating an API Key for an ELK Connector

When you are creating an ELK connection, you might need to generate an API key. Use the following procedure to generate the API Key.

Procedure

When you are completing these steps, replace the cluster and API key details with the details for your environment.

1. To check, or list, your existing **api_key**, run the following command:

```
curl -k -X GET 'https://was911.myibm.com:9200/_security/api_key' -H 'Authorization: Basic ZWxhc3RpYzpwYXNzdzByZA==' | jq '.'
```

2. If you need to create an **api_key**, run the following command:

```
curl -k -X POST 'https://was911.myibm.com:9200/_security/api_key' -H 'Content-Type: application/json' -H 'Authorization: Basic ZWxhc3RpYzpwYXNzdzByZA==' -d '{ "name": "my-api-key-sheikh-2", "role_descriptors": {} }' | jq '.'
```

3. Create the **encoded-base64-api-key** with the following command. Keep **"id": "LoRMmn4BCbCvoljFvMzq"** and **"api_key": "ZZEnRG93Q1i6wOCVW2XWug"**. These settings are needed to create the **encoded-base64-api-key**.

```
echo -n "{id}:{api_key}" | base64
```

For example, see the following example:

```
echo -n "LoRMmn4BCbCvoljFvMzq:ZZEnRG93Q1i6wOCVW2XWug" | base64
TG9STW1uNEJDYkN2b2xqRnZNenE6WlpFblJHOTNRMWk2d09DVlcyWFd1Zw==
```

4. Test the ELK query by using the **encoded-base64-api-key**:

```
curl -k -X 'GET' 'https://was911.myibm.com:9200/my-index' -H
'accept: /' -H 'Authorization: ApiKey {encoded-base64-api-key}}'
```

5. Test the key with the IBM Cloud Pak Automation console:

    1. Create an ELK connection with **api_key** as **encoded-base64-api-key**.
    2. Test the connection.
    3. Turn on data collection and save.
    4. Finally, ensure that your connection is saved.

Creating Elasticsearch, Fluentd, and Kibana (EFK) connections

You can connect to, and use, an Elasticsearch, Fluentd, and Kibana (EFK) stack that is deployed on Red Hat OpenShift as a log aggregator tool for IBM Cloud Pak for AIOps.

EFK is a suite of tools that functions as a log aggregation tool. Similar to an Elasticsearch, Logstash, and Kibana (ELK) connection with this type of connection, you can collect log data to establish a baseline of normal behavior and then identify anomalies. These anomalies can be correlated with other alerts and events, and published to your ChatOps interface to help you determine the cause and resolution of a problem.

You can connect to an Elasticsearch, Fluentd, and Kibana (EFK) stack as a variant option to connecting to an Elasticsearch, Logstash, and Kibana (ELK) stack.

To configure a connection to Elasticsearch, Fluentd, and Kibana (EFK), you can use the Red Hat OpenShift Logging library. The following instructions reference this library.

For more information about working with EFK connections, see the following sections:

- About this task
- Prerequisites
- Creating EFK connections
- Enabling EFK connections
- Editing EFK connections
- Deleting EFK connections
- Troubleshooting

About this task

Before creating the connection, you should be aware of the following information.

- **Load**: To prevent this connection placing an inordinate load on your data source and potentially impacting your logging operations, this connection only connects to one API with a default data frequency of 60 seconds. This is controlled by using the **Sampling rate** setting.
- **Access**: Custom data sources are cloud-based REST APIs. Access is configured by using the authentication methods that are specified in the **Authentication type** setting.
- **Data volume**: Data volume depends on the application, and is not a set value. Therefore, it does not appear in the settings.

Prerequisites

As a prerequisite, you need to make sure that you have Elasticsearch, Fluentd, and Kibana (EFK) set up on a cluster.

1. Install the Red Hat OpenShift Logging library. For more information, see the Red Hat OpenShift documentation Installing the logging subsystem for Red Hat OpenShift.

   When you are deploying the Red Hat OpenShift Logging (ClusterLogging) instance, use the following YAML definition as the base for your own definition.

   Change the values for the **storageClassName**, **maxAge**, and **size** depending on your own system. For example, if you are installing on Red Hat OpenShift on IBM Cloud (ROKS), use **ibmc-file-gold-gid** for the **storageClassName**. If you want to keep application logs for more than 15 days, update the **maxAge** value.

```
apiVersion: "logging.openshift.io/v1"
kind: "ClusterLogging"
metadata:
  name: "instance"
  namespace: "openshift-logging"
spec:
  managementState: "Managed"
  logStore:
    type: "elasticsearch"
    retentionPolicy:
      application:
        maxAge: 15d
    elasticsearch:
      nodeCount: 3
      storage:
        storageClassName: "rook-cephfs"
        size: 80G
      resources:
        limits:
            memory: "16Gi"
        requests:
            memory: "16Gi"
      proxy:
        resources:
          limits:
            memory: 256Mi
          requests:
            memory: 256Mi
      redundancyPolicy: "SingleRedundancy"
  visualization:
    type: "kibana"
    kibana:
      replicas: 1
  collection:
    logs:
      type: "fluentd"
      fluentd: {}
```

2. Verify that all pods under the **openshift-logging** namespace are running by running the following command.

```
oc get pods -n openshift-logging
```

   If your pods are not running, it can take several minutes for all pods to be in the **Running** state. For instance, the **collector** pods can go through several restarts before they are all running. You can have several pods for ClusterLogging, Elasticsearch, fluentd (**collector**), and Kibana similar to the example output:

```
NAME                                            READY   STATUS    RESTARTS   AGE
cluster-logging-operator-84854b544c-tc758       1/1     Running   0          26m
collector-5hndx                                 2/2     Running   0          10m
collector-6sfk9                                 2/2     Running   0          10m
collector-f5w84                                 2/2     Running   0          10m
collector-fgzsr                                 2/2     Running   0          10m
collector-fhsbk                                 2/2     Running   0          10m
collector-fwdhk                                 2/2     Running   0          10m
collector-hj2q9                                 2/2     Running   0          10m
collector-l4v5q                                 2/2     Running   0          10m
elasticsearch-cdm-medm3kgq-1-74fb867496-swgwx   2/2     Running   0          22m
elasticsearch-cdm-medm3kgq-2-6cb7495ff5-h5bgd   2/2     Running   0          22m
elasticsearch-cdm-medm3kgq-3-5fb8f9f4-bgqgp     2/2     Running   0          22m
```

```
elasticsearch-im-app-27288900-b8j28                0/1     Completed   0          2m3s
elasticsearch-im-audit-27288900-btw2x              0/1     Completed   0          2m3s
elasticsearch-im-infra-27288900-ndjcb              0/1     Completed   0          2m3s
kibana-5586499d9b-45x7c                            2/2     Running     0          22m
```

3. **Optional:** If you intend to limit the scope to a particular namespace in your cluster, you can complete the following steps:

    1. Update the ClusterLogging instance `managementState` field from `Managed` to `Unmanaged`. This change can give you more control of the components that are managed by the Red Hat OpenShift Logging Operator. To change the setting, run the following command to edit the instance:

        ```
        oc edit clusterlogging instance -n openshift-logging
        ```

    2. Update the `path` variable in the `collector` configmap under the container logs section.

        You can use the following command to update the configmap.

        ```
        oc edit configmap collector -n openshift-logging
        ```

        As an alternative, you can also use the Red Hat OpenShift Console. To use the console, go to **Workloads** > **ConfigMaps**. Then, from the **Projects** drop-down, select *openshift-logging*. Select the `collector` configmap and edit the YAML. By default, this variable has the following value in the `collector` configmap:

        ```
        path "/var/log/pods/*/*/*.log"
        ```

        Change the value to reference your preferred namespace, for example:

        ```
        path "/var/log/pods/qotd_qotd-*/*/*.log"
        ```

    3. If you updated the collector `ConfigMapRestart` then restart all collector pods (daemonset):

        ```
        oc get pods -n openshift-logging --no-headers=true | awk '/collector/{print $1}' |
        xargs  oc delete -n openshift-logging pod
        ```

4. Expose the log store service as a route. For more information, see the Red Hat OpenShift documentation Exposing the log store service as a route.

    When you are completing this procedure, obtain the token for the `cluster-logging-operator` service account instead of your own Red Hat OpenShift Container Platform token. You must be logged in to the cluster where you installed the `ClusterLogging` instance.

    ```
    token=$(oc sa get-token cluster-logging-operator -n openshift-logging)
    echo $token
    ```

5. When the token variable is set, run the following commands on your workstation to validate that you can connect to Elasticsearch:

    ```
    routeES=`oc get route elasticsearch -o jsonpath={.spec.host} -n openshift-logging`
    curl -tlsv1.2 --insecure -H "Authorization: Bearer ${token}" "https://${routeES}"
    curl -tlsv1.2 --insecure -H "Authorization: Bearer ${token}" "https://${routeES}/_search?
    scroll=1m"
    ```

    If needed, you can run the following command to get the Elasticsearch version:

    ```
    curl --insecure -H "Authorization: Bearer ${token}" https://${routeES}/
    ```

6. Index the Elasticsearch documents. This indexing is done automatically, but the process can take a few minutes in a new or updated cluster. Wait a few minutes for the changes to display in the Kibana UI.

    When all configuration steps take effect, you can see entries in the Kibana UI for only the application that you intend to collect logs from.

Creating EFK connections

With EFK set up, you can now create the connection in IBM Cloud Pak for AIOps.

For this connection you need to select the tile for creating an Elasticsearch, Logstash, and Kibana (ELK) connection, but change some settings to connect to EFK instead of ELK.

To create the connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.
4. From the list of available connections, find and click the **ELK** tile.

   Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Connect**.
6. On the *Add connection* page, define the general connection details:

   - **Name**: The display name of your connection.
   - **Description**: An optional description for the connection.
   - **ELK service URL**: ELK service URL: The Elasticsearch host and public API port. The URL must have the target index that IBM Cloud Pak for AIOps uses to search for the data from your applications. If you want to use all data, specify **`*`** as your index. For example, your ELK service URL can be **`https://myURL.com:8080/*`**.

     Run the following command to find the hostname for the URL:

     ```
     oc get routes -n openshift-logging | grep elastic
     ```

     Example output:

     ```
     elasticsearch    elasticsearch-openshift-logging.apps.demo-apps.cp.mysite.com
     elasticsearch    <all>   reencrypt    None
     ```

     When you have your Elastic hostname, you can build your URL. For example, **`https://elasticsearch-openshift-logging.apps.demo-apps.cp.mysite.com/app*`**. Ensure that you also replace **`/app*`** in the URL with the value for your EFK pattern.
   - **Kibana URL**: Enter a URL for the service instance. Run the following command to find the hostname for the URL:

     ```
     oc get routes -n openshift-logging | grep kibana
     ```
   - **Authentication type**: Set this value to *Token* to indicate the Elasticsearch instance is authenticated with a temporary token.

     For the token, run the following command to obtain the token for the **`cluster-logging-operator`** service account:

     ```
     token=$(oc sa get-token elasticsearch -n openshift-logging)
     echo $token
     ```
   - **Certificate (optional)**: Certificate used to verify the SSL/TLS connection to the REST service.
   - **Filters (optional)**: A custom Boolean Query to filter the Elasticsearch request for your specific application, terms, keywords, or other filters.
   - **Time zone (optional)**: The time zone in which your data is situated. The default time is converted from the system time relative to UTC time. The default value is UTC.
   - **Kibana port:* The port of the Kibana instance that is on the same host as the Elasticsearch instance. Use **`443`** for this field.
   - **Base parallelism**: Select a value to specify the number of Flink jobs that can run in parallel. These jobs run to process and normalize the collected data. The default value is 1. However, it is recommended to use a higher value than 1 so that you can process data in parallel. This value cannot exceed the total available free Flink slots.

In a small environment, the available flinks slots are 16, while in a large environment, the maximum available slots are 32. If you are collecting historical data with this connection, you can set this value to be equal to the **source** parallelism.

- **Sampling rate**: The rate at which data is pulled from live source (in seconds). The default value is `60`.
- **JSON processing option**: Select a JSON processing option.
  - **None**: The default option. The JSON is not processed or modified.
  - **Flatten**: This option flattens the JSON object by removing the opening and closing braces.
  - **Filter**: This option extracts the JSON object and replaces it with an empty string.
  - For more information about the options, see Managing embedded JSON.

Note: To improve data throughput, you can increase the base parallelism value incrementally. For more information about maximum base parallelism for starter and production deployments, see Improving data streaming performance for log anomaly detection.

Note: If you use the Filter option, do not use a timestamp in the filter query. This setting can cause a parsing error in the backend.

```
    "range": {
      "@timestamp": {
        "gte": "now-2m",
        "lt" : "now"
      }
```

Other than that limitation, filter can use any clauses so long as the fields and the values specified in the filter are relevant to the target endpoint data set.

7. You can test your connection by clicking **Test connection**.
8. Click **Next**.
9. Enter *Field Mapping* information (Optional):

   You can improve search performance by mapping the fields from your implementation fields to IBM Cloud Pak for AIOps's standard fields.

   Use the following mapping instead of the default mapping that is provided for an ELK connection:

```
{
  "codec": "elk",
  "message_field": "message",
  "log_entity_types": "kubernetes.container_image_id, kubernetes.host,
kubernetes.pod_name, kubernetes.namespace_name",
  "instance_id_field": "kubernetes.container_name",
  "rolling_time": 10,
  "timestamp_field": "@timestamp"
}
```

10. Click **Next**.
11. Enter AI training and log data (Optional):

    Select how you want to manage collecting data for use in AI training and anomaly detection. Click the **Data collection** toggle to turn on data collection, then select how you want to collect data:

    - **Live data for continuous AI training and anomaly detection**: A continuous collection of data from your connection is used to both train AI models and analyze your data for anomalous behavior.

      Note: After an initial installation, there is no data at all in the system. If you select this option, then the two different log anomaly detection algorithms behave in the following ways:

      - Natural language log anomaly detection does not initially detect anomalies as no model has been trained. You can retrieve historical data (select **Historical data for initial AI training**) to speed up the retrieval of data to train on, or you can leave the **Live data for continuous AI training and anomaly detection** setting on. In the latter case, the system gathers training data live and after a few days there is enough data to train a model. When this model is deployed, then it detects anomalies as normal.

- Statistical baseline log anomaly detection does not detect anomalies for the first 30 minutes of data collection. This is because it does not have a baseline yet. After 30 minutes of live data collection the baseline is automatically created. After that it detects anomalies on an ongoing basis, while continuing to gather data and improve its model every 30 minutes.

- **Live data for initial AI training**: A single set of training data used to define your AI model. Data collection takes place over a specified time period that starts when you create your connection.

  Note: Selecting this option causes the system to continue to collect data while the option is enabled; however, the data is collected for training only, and not for log anomaly detection. For more information about AI model training, including minimum and ideal data quantities, see Configuring AI training.

- **Historical data for initial AI training**: A single set of training data used to define your AI model. You need to give Start and End dates, and specify the parallelism of your source data. Historical data is harvested from existing logs in your connection over a specified time period in the past.

  - **Start date**: Select a start date from the calendar and enter the time in hh:mm (hours and minutes) format.

    Note: The start date must not exceed 31 days from the present as the maximum time period for historical data collection is 31 days. The recommended time period is two weeks.

  - **Time zone**: Select your time zone from the dropdown list.

  - **End date and time**: Click **Add end date** and select an end date from the calendar and enter the time in hh:mm format.

    Note: If you do not specify the end date, then live data collection follows the historical data collection. If you do not want to set an end date, click **Remove end date**.

  - **Source parallelism (1-50)**: Select a value to specify the number of requests that can run in parallel to collect data from the source. Generally, you can set the value to equal the number of days of datat that you want to collect. When you are setting this value, consider the number of requests that are allowed by the source in a minute. For example, if only 1-2 requests are allowed, set the value to be low.

Important: Keep in mind the following considerations when you select your data collection type:

- Anomaly detection for your connection occurs if you select *Live data for continuous AI training and anomaly detection*.
- Different types of AI models have different requirements to properly train a model. Make sure that your settings satisfy minimum data requirements. For more information about how much data you need to train different AI models, see Configuring AI training.

12. Click **Done**.

You created the connection in your instance. After you create your connection, you must enable the data collection to connect your connection with the AI of IBM Cloud Pak for AIOps.

Enabling and disabling EFK connections

If you didn't enable your data collection during creation, you can enable your connection afterward. You can also disable a previously enabled connection the same way. If you selected *Live data for initial AI training* when you created your connection, you must disable the connection before AI model training. To enable or disable a created connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Manage connections* tab of the *Data and tool connections* page, click the **ELK** connection type.
4. Click the connection that you want to enable or disable.
5. Go to the *AI training and log data* section. Set **Data connection** to **On** or **Off** to enable or disable data collection. Disabling data collection for a connection does not delete the connection.

You enabled or disabled your connection. For more information about deleting a connection, see Deleting EFK connections.

Editing EFK connections

After you create your connection, your can edit the connection. For example, if you specified *Historical data for initial AI training* but now want your connection to pull in live data for continuous monitoring, you can edit it. To edit a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **ELK** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the ELK connections page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.
5. Edit your connection as required. Click **Save** when you are done editing.

Your connection is now edited. If your application was not previously enabled or disabled, you can enable or disable the connection directly from the interface. For more information about enabling and disabling your connection, see Enabling and disabling EFK connections. For more information about deleting a connection, see Deleting EfK connections.

Deleting EFK connections

If you no longer need your EFK connection and want to not only disable it, but delete it entirely, you can delete the connection from the console.

Note: You must disable data collection before you delete your connection. For more information about disabling data collection, see Enabling and disabling EFK connections.

To delete a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **ELK** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the ELK connections page, click the options menu (three vertical dots) for the connection that you want to delete and click **Delete**.
5. Enter the name of the connection to confirm that you want to delete your connection. Then, click **Delete**.

Your connection is deleted.

Creating and managing Falcon LogScale connections

A Falcon LogScale (previously named Humio) connection provides log data, which is used to establish a baseline of normal behavior and then identify anomalies. These anomalies can be correlated with other alerts and events, and published to your ChatOps interface, to help you determine the cause and resolution of a problem.

For more information about working with Falcon LogScale connections, see the following sections:

- Creating Falcon LogScale connections
- Enabling Falcon LogScale connections
- Editing Falcon LogScale connections
- Deleting Falcon LogScale connections

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating Falcon LogScale connections

About this task

Before creating the connection, you should be aware of the following information.

- **Load**: To prevent this connection placing an inordinate load on your data source and potentially impacting your logging operations, this connection only connects to one API with a default data frequency of 60 seconds. This is controlled by using the **Sampling rate** setting in the Procedure section.
- **Access**: Custom data sources are cloud-based REST APIs. Access is configured by using the authentication methods that are specified in the **Authentication type** setting in the Procedure section.
- **Data volume**: Data volume depends on the application, and is not a set value. Therefore, it does not appear in the settings.

Procedure

To create a custom connection from a specific source, step through the following sections:

- Adding a connection
- Specifying connection parameters
- Specifying field mapping
- Specifying how log data is collected for AI training

Adding a connection

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.
4. From the list of available connections, find and click the **Falcon LogScale** tile.

   Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Connect**.
6. On the *Add connection* page, define the general connection details:

   - **Name**: The display name of your connection.
   - **Description**: An optional description for the connection.
   - **Falcon LogScale service URL**: The fully qualified Falcon LogScale query API URL for your Falcon LogScale instance, for example: `http://<Falcon LogScale_HOSTNAME>/api/v1/repositories/<REPO_NAME>/query`
   - **API Token**: The connection token for your Falcon LogScale instance, which is available from the account settings page in Falcon LogScale.
   - **Certificate**: An optional certificate used to verify the SSL/TLS connection to the REST service.
   - **Filters**: Optional filters that define subsets of data that is pulled from Falcon LogScale.
   - **Base parallelism**: Select a value to specify the number of Flink jobs that can run in parallel. These jobs run to process and normalize the collected data. The default value is 1. However, it is recommended to use a higher value than 1 so that you can process data in parallel. This value cannot exceed the total available free Flink slots. In a small environment, the available flinks slots are 16, while in a large environment, the maximum available slots are 32. If you are collecting historical data with this connection, you can set this value to be equal to the **source** parallelism.
   - **Sampling rate**: The rate at which data is pulled from live source (in seconds). The default value is `60`.
   - **JSON processing option**: Select a JSON processing option.
     - **None**: The default option. The JSON is not processed or modified.
     - **Flatten**: This option flattens the JSON object by removing the opening and closing braces.
     - **Filter**: This option extracts the JSON object and replaces it with an empty string.
     - For more information about the options, see Managing embedded JSON.

Note: To improve data throughput, you can increase the base parallelism value incrementally. For more information about maximum base parallelism for starter and production deployment sizes, see Improving data streaming performance for log anomaly detection.

7. Test your connection by clicking **Test connection**.
8. Click **Next** to move to the next page.

Specifying field mapping

1. Enter **Field Mapping** information (Optional):

   The default mapping setting that is provided by UI can improve the search performance by mapping the fields from your implementation fields to IBM Cloud Pak for AIOps's standard fields. For more information about how field mappings are defined, see Mapping data from incoming sources. For more information about the use of mappings to clean your data for use in IBM Cloud Pak for AIOps, see Cleaning mapped data that use regular expressions. Consider the supported data schema when you create your field mapping:

   ```
   {
     "codec": "humio",
     "message_field": "@rawstring",
     "log_entity_types": "kubernetes.container_name",
     "instance_id_field": "kubernetes.container_name",
     "rolling_time": 10,
     "timestamp_field": "@timestamp",
     "resource_id": "kubernetes.pod_name"
   }
   ```

   But processing of logs with this format into log anomaly alerts can lead to false alarms in the Alert Viewer.

2. To resolve the false alarms issue in step 1, use regular expression to exclude parts of the log lines at the end of fields mapping (Optional):

   Logs ingested by Falcon LogScale usually have the following format:

   - `_timestamp_ + stdout F + _real message_`
   - `_timestamp_ + stderr F + _real message_`

   You can define the field mapping as follows:

   ```
   {
   "codec": "humio",
   "message_field": "@rawstring",
   "log_entity_types": "kubernetes.container_name",
   "instance_id_field": "kubernetes.container_name",
   "rolling_time": 10,
   "timestamp_field": "@timestamp",
      "custom_regex": [
      ".+?(stdout F )",
      ".+?(stderr F )"
      ]
   }
   ```

3. Click **Next** to move to the next page.

Specifying how log data is collected for AI training

1. Enter *AI training and log data* (Optional):

   Select how you want to manage collecting data for use in AI training and anomaly detection. Set the **Data collection** toggle to 'on', then select how you want to collect data:

   - **Live data for continuous AI training and anomaly detection**: A continuous collection of data from your connection is used to both train AI models and analyze your data for anomalous behavior.

     Note: After an initial installation, there is no data at all in the system. If you select this option, then the two different log anomaly detection algorithms behave in the following ways:

- Natural language log anomaly detection does not initially detect anomalies as no model has been trained. You can retrieve historical data (select **Historical data for initial AI training**) to speed up the retrieval of data to train on, or you can leave the **Live data for continuous AI training and anomaly detection** setting on. In the latter case, the system gathers training data live and after a few days there is enough data to train a model. When this model is deployed, then it detects anomalies as normal.

- Statistical baseline log anomaly detection does not detect anomalies for the first 30 minutes of data collection. This is because it does not have a baseline yet. After 30 minutes of live data collection the baseline is automatically created. After that it detects anomalies on an ongoing basis, while continuing to gather data and improve its model every 30 minutes.

- **Live data for initial AI training**: A single set of training data used to define your AI model. Data collection takes place over a specified time period that starts when you create your connection.

  Note: Selecting this option causes the system to continue to collect data while the option is enabled; however, the data is collected for training only, and not for log anomaly detection. For more information about AI model training, including minimum and ideal data quantities, see Configuring AI training.

- **Historical data for initial AI training**: A single set of training data used to define your AI model. You need to give Start and End dates, and specify the parallelism of your source data. Historical data is harvested from existing logs in your connection over a specified time period in the past.

  - **Start date**: Select a start date from the calendar and enter the time in hh:mm (hours and minutes) format.

    Note: The start date must not exceed 31 days from the present as the maximum time period for historical data collection is 31 days. The recommended time period is two weeks.

  - **Time zone**: Select your time zone from the dropdown list.

  - **End date and time**: Click **Add end date** and select an end date from the calendar and enter the time in hh:mm format.

    Note: If you do not specify the end date, then live data collection follows the historical data collection. If you do not want to set an end date, click **Remove end date**.

  - **Source parallelism (1-50)**: Select a value to specify the number of requests that can run in parallel to collect data from the source. Generally, you can set the value to equal the number of days of datat that you want to collect. When you are setting this value, consider the number of requests that are allowed by the source in a minute. For example, if only 1-2 requests are allowed, set the value to be low.

    Note: To avoid issues with task manager memory, run no more than one *Historical data for initial AI training* run at a time.

Important: Keep in mind the following considerations when you select your data collection type:

- Anomaly detection for your connection occurs if you select *Live data for continuous AI training and anomaly detection*.

- Different types of AI models have different requirements to properly train a model. Make sure that your settings satisfy minimum data requirements. For more information about how much data you need to train different AI models, see Configuring AI training.

2. Click **Done**.

You have created a Falcon LogScale connection in your instance. After you create your connection, you must enable the data collection to connect your connection with the AI of IBM Cloud Pak for AIOps. For more information about enabling your connection, see Enabling Falcon LogScale connections.

To create more connections (such as a ChatOps connection), see Configuring data and tool connections.

For more information about working with the insights provided by your connections, see ChatOps insight management.

Enabling and disabling Falcon LogScale connections

If you didn't enable your data collection during creation, you can enable your connection afterward. You can also disable a previously enabled connection the same way. If you selected *Live data for initial AI training* when you created your connection, you must disable the connection before AI model training. To enable or disable a created connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Manage connections* tab of the *Data and tool connections* page, click the **Falcon LogScale** connection type.
4. Click the connection that you want to enable or disable.
5. Go to the *AI training and log data* section. Set **Data connection** to **On** or **Off** to enable or disable data collection. Disabling data collection for a connection does not delete the connection.

You have enabled or disabled your connection. For more information about deleting a connection, see Deleting Falcon LogScale connections.

Editing Falcon LogScale connections

After you create your connection, your can edit the connection. For example, if you specified *Historical data for initial AI training* but now want your connection to pull in live data for continuous monitoring, you can edit it. To edit a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Falcon LogScale** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the Falcon LogScale connections page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.
5. Edit your connection as required. Click **Save** when you are done editing.

Your connection is now edited. If your application was not previously enabled or disabled, you can enable or disable the connection directly from the interface. For more information about enabling and disabling your connection, see Enabling and disabling Falcon LogScale connections. For more information about deleting a connection, see Deleting Falcon LogScale connections.

Deleting Falcon LogScale connections

If you no longer need your Falcon LogScale connection and want to not only disable it, but delete it entirely, you can delete the connection from the console.

Note: You must disable data collection before deleting your connection. For more information about disabling data collection, see Enabling and disabling Falcon LogScale connections.

To delete a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Falcon LogScale** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the Falcon LogScale connections page, click the options menu (three vertical dots) for the connection that you want to delete and click **Delete**.
5. Enter the name of the connection to confirm that you want to delete your connection. Then, click **Delete**.

Your connection is deleted.

Creating Generic Webhook connections

If you want to connect to a data source and a dedicated connection is not available for that source in the IBM Cloud Pak for AIOps Data and tools connection UI, you can use the generic webhook connection. This connection provides an endpoint for applications to send JSON formatted events to IBM Cloud Pak for AIOps through HTTP POST requests.

In the requests you can also include JSONata that maps incoming JSON payloads to IBM Cloud Pak for AIOps events.

For more information about working with a generic webhook connections, see the following sections:

- Before you begin
- Creating a Generic Webhook connection
- Editing a Generic Webhook connection
- Deleting a Generic Webhook connection

Before you begin

To configure a generic webhook connection, you need the following information for the application that you are connecting to IBM Cloud Pak for AIOps:

- A name and description for the webhook connection.
- If you need to secure the webhook, the appropriate user name and password.
- The JSONata that maps the incoming JSON payload to IBM Cloud Pak for AIOps events.

Creating a Generic Webhook connection

To create a Generic Webhook connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.
4. From the list of available connections, find and click the **Generic Webhook** tile.

   Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section (Events).
5. On the side-panel, review the instructions and when ready to continue, click **Connect**.
6. On the *Add connection* page, define the general connection details:
   - **Name**: The display name of your connection. You cannot use spaces or special characters in the display name. Use only alphanumeric characters.
   - **Description**: An optional description for the connection.
   - **Authentication type**: Select the **Username/Password** or **None** option.
   - **Username**: Enter the appropriate username to authenticate the connection.
   - **Password**: Enter the appropriate password to authenticate the connection.
   - **Webhook route**: The predefined full URL (location) of the webhook. This field is not editable.
7. Click **Next**.
8. Configure the event mapping.

   Use JSONATA to map the incoming JSON payload to the IBM Cloud Pak for AIOps event data schema. You can start from scratch or load an example mapping. The mapping configuration will be tested to ensure that it is in a valid

JSONata format.

The following mapping is an example mapping for a connection with IBM SevOne Network Management System (NMS)

```
(
    {
        "severity": alertState="Alert"?6:alertState="Emergency"?6:alertState="Critical"?
6:alertState="Error"?5:alertState="Warning"?3:alertState="Notice"?2:alertState="Info"?
2:alertState="Debug"?2:1,
        "summary": description,
        "resource": {
            "name": deviceName
        },
        "type": {
            "classification": policyName,
            "eventType":  alertState = "Cleared" ? "resolution" : "problem"
        },
        "sender": {
            "name": "IBM SevOne Network Management System",
            "type": "Webhook Connector"
        },
        "expirySeconds": alertId="-1"?300
    }
)
```

Note: The JSONata expression must contain the `summary`, `resource` and `type` fields. If the fields are missing, the event is invalid and might be rejected. The connector assigns the following default values to the event data:

- `"summary":"Unknown event."`
- `"resource":{"name":"Unknown"}`
- `"type":{"eventType":"problem","classification":null,"condition":null}`

Review the following topics for more example mappings to help you configure your own mappings:

- Example: IBM SevOne Network Management System (NMS) generic webhook event mappings
- Example: Prometheus AlertManager (Red Hat OpenShift) generic webhook event mappings

9. Enable the webhook. Click the toggle to change the setting **Enable webhook** or **Disable webhook**. When disabled, the webhook connector discards incoming HTTP requests.

10. Click **Done**.

11. Within the application that you are connecting to IBM Cloud Pak for AIOps, complete any required configurations to support the connection.

    Important: You might need to configure a signed trusted certificate for IBM Cloud Pak for AIOps that the external system that you are connecting to trusts. When an external system is sending a payload through this webhook connection, the payload goes through a TLS handshake. If IBM Cloud Pak for AIOps is using the default configuration, IBM Cloud Pak for AIOps uses self-signed certificates that an external system will not accept if the system does not trust self-signed certificates. To configure a signed trusted certificate, If you are connecting to a SaaS system, you might need to use an internationally recognised certificate authority certificate as you might need be able to update the SaaS trusted certificate authority list. Otherwise, you can use your own certificate authority and add the certificate to the external system.

    For example, to connect to IBM SevOne Network Management System (NMS) you need to complete the following tasks:

    - Copy the Webhook URL from IBM Cloud Pak for AIOps.
    - Importing a CA signer certificate into IBM SevOne Network Management System (NMS).
    - Add the webhook destination within IBM SevOne Network Management System (NMS).

      In the IBM SevOne Network Management System (NMS) UI go to **Events** > **Configuration** > **Policy Browser** > **Configure webhook destination** > **Add webhook destination**.

    - Add the webhook into existing policies to send event data to IBM Cloud Pak for AIOps.

12. Test your connection with your connected application to verify that events are sent to IBM Cloud Pak for AIOps, and display as alerts in the Alert Viewer of the IBM Cloud Pak for AIOps UI.

Enabling or disabling a Generic Webhook connection

After you create your webhook connection, you can enable or disable the webhook by completing the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Generic Webhook** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the *Generic Webhook* page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.
5. Enable or disable the webhook by clicking the toggle to change the setting to **Enable webhook** or **Disable webhook**. When disabled, the webhook connector discards incoming HTTP requests.
6. Click **Save** when you are done.

Your webhook is now enabled or disabled.

Editing a Generic Webhook connection

After you create your connection, you can edit the connection. For example, if you want to add or change the listed GUI or backend URLs for the IBM Tivoli Netcool/Impact endpoint, or to update the User ID or password. To edit a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Generic Webhook** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the *Generic Webhook* page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.
5. Edit your connection as required. Click **Save** when you are done editing.

Your connection is now edited.

Deleting a Generic Webhook connection

If you no longer need your connection and want to delete it entirely, you can delete the connection from the console.

To delete a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Generic Webhook** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the *Generic Webhook* connections page, click the *Delete* icon for the connection that you want to delete.
5. Enter the name of the connection to confirm that you want to delete your connection. Then, click **Delete**.

Your connection is deleted.

Example: IBM SevOne Network Management System (NMS) generic webhook event mappings

Review the following example mappings for a generic webhook connection between IBM Cloud Pak for AIOps and IBM SevOne Network Management System (NMS). Use these mappings to help you configure your own event mapping for a generic webhook connection.

- Example event (alert) JSON
- Example event mapping JSONata expression
- Example event result JSON
- Example Curl Command

Example event (alert) JSON


```
{
    "routes": [
        "IBM CloudPak for AIOps"
    ],
    "host": "hur-ka-cpb2a.net.hursley.ibm.com",
    "description": "Threshold triggered -- hur-ka-cpb2a.net.hursley.ibm.com's Ethernet1/59's
Out Discards: 147.98 Number >= 120 Number totaled over 5.00 minutes",
    "alertMessage": "Threshold triggered -- hur-ka-cpb2a.net.hursley.ibm.com's Ethernet1/59's
Out Discards: 147.98 Number >= 120 Number totaled over 5.00 minutes",
    "check": "AS-4.3-Nov - Ethernet1/59",
    "cluster": "Hursley-A",
    "alertId": "913284",
    "alertType": "system",
    "alertState": "Critical",
    "occurrences": "1",
    "assignedTo": "unassigned",
    "deviceId": "42",
    "deviceIp": "9.20.253.60",
    "deviceName": "hur-ka-cpb2a.net.hursley.ibm.com",
    "deviceAltName": "",
    "groupName": "Special",
    "objectId": "17325",
    "objectName": "Ethernet1/59",
    "objectAltName": "",
    "objectDescription": "fcxhurb14",
    "pluginName": "SNMP",
    "pluginDescription": "SNMP Poller",
    "policyId": "84",
    "policyName": "AS-4.3-Nov",
    "thresholdId": "39649",
    "thresholdName": "AS-4.3-Nov - hur-ka-cpb2a.net.hursley.ibm.com - Ethernet1/59",
    "linkURL": "<DI_URL.com/redirect/v1/reports?
reportName=Indicator%20Summary&datasourceName=Hursley-
A&indicators=SNMP%26%26Interface%26%26hur-ka-
cpb2a.net.hursley.ibm.com%26%26Ethernet1%2F59%26%26ifOutDiscards&startTime=1701075605000|View
in Data Insight>",
    "triggeringConditions": [
        {
            "aggregationDuration": "5.00",
            "aggregationOperation": "Total",
            "baselineValue": "0.00",
            "comparisonOperation": ">=",
            "comparisonUnits": "Number",
            "comparisonValue": "120",
            "dataUnits": "Number",
            "dataValue": "147.98",
            "indicatorDescription": "Out Discards",
            "indicatorName": "ifOutDiscards",
            "sigmaDirection": "above or below",
            "sigmaValue": "nan",
            "thresholdValue": "120.00"
        }
    ],
    "closureMessage": "N/A"
}
```


Example event mapping JSONata expression

```
(
    {
        "severity": alertState="Alert" ? 6 : alertState="Emergency" ? 6 :
alertState="Critical" ? 6 : alertState="Error" ? 5 : alertState="Warning"? 3 :
alertState="Notice" ? 2 : alertState="Info"? 2 : alertState="Debug" ? 2 : 1,
        "summary": description,
        "resource": {
            "name": deviceName
        },
        "type": {
            "classification": policyName,
            "eventType": ($exists(closureMessage) and closureMessage != "N/A") ? "resolution"
: alertState = "Cleared" ? "resolution" : "problem"
        },
        "sender": {
            "name": "IBM SevOne",
            "type": "Webhook Connector"
        },
        "expirySeconds": alertId="-1" ? 300
    }
)
```

Example event result JSON

```
{
    "summary": "Threshold triggered -- hur-ka-cpb2a.net.hursley.ibm.com's Ethernet1/59's Out
Discards: 147.98 Number >= 120 Number totaled over 5.00 minutes",
    "severity": 6,
    "sender": {
        "name": "IBM SevOne",
        "type": "Webhook Connector"
    },
    "resource": {
        "name": "hur-ka-cpb2a.net.hursley.ibm.com"
    },
    "occurrenceTime": 1701662721822,
    "type": {
        "condition": null,
        "eventType": "problem",
        "classification": "AS-4.3-Nov"
    }
}
```

Example Curl Command

1. Create a payload json file

```
cat << EOF | tee > sevone-payload.json | cat
  {
    "routes": [
        "IBM CloudPak for AIOps"
    ],
    "host": "hur-ka-cpb2a.net.hursley.ibm.com",
    "description": "Threshold triggered -- hur-ka-cpb2a.net.hursley.ibm.com's Ethernet1/59's
Out Discards: 147.98 Number >= 120 Number totaled over 5.00 minutes",
    "alertMessage": "Threshold triggered -- hur-ka-cpb2a.net.hursley.ibm.com's Ethernet1/59's
Out Discards: 147.98 Number >= 120 Number totaled over 5.00 minutes",
    "check": "AS-4.3-Nov - Ethernet1/59",
    "cluster": "Hursley-A",
    "alertId": "913284",
    "alertType": "system",
    "alertState": "Critical",
    "occurrences": "1",
    "assignedTo": "unassigned",
    "deviceId": "42",
    "deviceIp": "9.20.253.60",
    "deviceName": "hur-ka-cpb2a.net.hursley.ibm.com",
    "deviceAltName": "",
    "groupName": "Special",
```

```
    "objectId": "17325",
    "objectName": "Ethernet1/59",
    "objectAltName": "",
    "objectDescription": "fcxhurb14",
    "pluginName": "SNMP",
    "pluginDescription": "SNMP Poller",
    "policyId": "84",
    "policyName": "AS-4.3-Nov",
    "thresholdId": "39649",
    "thresholdName": "AS-4.3-Nov - hur-ka-cpb2a.net.hursley.ibm.com - Ethernet1/59",
    "linkURL": "<DI_URL.com/redirect/v1/reports?
reportName=Indicator%20Summary&datasourceName=Hursley-
A&indicators=SNMP%26%26Interface%26%26hur-ka-
cpb2a.net.hursley.ibm.com%26%26Ethernet1%2F59%26%26ifOutDiscards&startTime=1701075605000|View
in Data Insight>",
    "triggeringConditions": [
        {
            "aggregationDuration": "5.00",
            "aggregationOperation": "Total",
            "baselineValue": "0.00",
            "comparisonOperation": ">=",
            "comparisonUnits": "Number",
            "comparisonValue": "120",
            "dataUnits": "Number",
            "dataValue": "147.98",
            "indicatorDescription": "Out Discards",
            "indicatorName": "ifOutDiscards",
            "sigmaDirection": "above or below",
            "sigmaValue": "nan",
            "thresholdValue": "120.00"
        }
    ],
    "closureMessage": "N/A"
  }
EOF

cat << EOF | tee > sevone-payload-clear.json | cat
{
  "routes": ["IBM CloudPak for AIOps"],
  "host": "hur-ka-cpb2a.net.hursley.ibm.com",
  "description": "Threshold triggered -- hur-ka-cpb2a.net.hursley.ibm.com's Ethernet1/29's Out
Discards: 800.34 Number >= 100 Number totaled over 3.00 minutes",
  "alertMessage": "Threshold triggered -- hur-ka-cpb2a.net.hursley.ibm.com's Ethernet1/29's
Out Discards: 800.34 Number >= 100 Number totaled over 3.00 minutes",
  "check": "AS-4.3-Nov - Ethernet1/29",
  "cluster": "Hursley-A",
  "alertId": "898689",
  "alertType": "system",
  "alertState": "Critical",
  "occurrences": "2",
  "assignedTo": "unassigned",
  "deviceId": "42",
  "deviceIp": "9.20.253.60",
  "deviceName": "hur-ka-cpb2a.net.hursley.ibm.com",
  "deviceAltName": "",
  "groupName": "Special",
  "objectId": "17296",
  "objectName": "Ethernet1/29",
  "objectAltName": "",
  "objectDescription": "fshurb1",
  "pluginName": "SNMP",
  "pluginDescription": "SNMP Poller",
  "policyId": "84",
  "policyName": "AS-4.3-Nov",
  "thresholdId": "39640",
  "thresholdName": "AS-4.3-Nov - hur-ka-cpb2a.net.hursley.ibm.com - Ethernet1/29",
  "linkURL" : "<DI_URL.com/redirect/v1/reports?
reportName=Indicator%20Summary&datasourceName=Hursley-
A&indicators=SNMP%26%26Interface%26%26hur-ka-
cpb2a.net.hursley.ibm.com%26%26Ethernet1%2F29%26%26ifInErrors%7C%7CSNMP%26%26Interface%26%26hu
r-ka-
cpb2a.net.hursley.ibm.com%26%26Ethernet1%2F29%26%26ifOutErrors%7C%7CSNMP%26%26Interface%26%26h
```

```
ur-ka-
cpb2a.net.hursley.ibm.com%26%26Ethernet1%2F29%26%26ifInDiscards%7C%7CSNMP%26%26Interface%26%26
hur-ka-
cpb2a.net.hursley.ibm.com%26%26Ethernet1%2F29%26%26ifOutDiscards&startTime=1700650083000|View
in Data Insight>",
  "triggeringConditions": [] ,
  "closureMessage": "Alert cleared by clear condition"
}
EOF
```

2. Send the payload with a `curl` command

```
# Webhook details
WEBHOOK_URL=https://webhook-connector-1a2bc34d-1ab2-1abc-1a2b-1234a56bc789-
aiops.apps.mysite.com/webhook-connector/abcdefgh1jk
AUTH=test:test

# Send alert
curl -X POST -u ${AUTH} --insecure \
-H 'Content-Type: application/json' ${WEBHOOK_URL} \
-d @./sevone-payload.json


# Send clear event
curl -X POST -u ${AUTH} --insecure \
-H 'Content-Type: application/json' ${WEBHOOK_URL} \
-d @./sevone-payload-clear.json
```

Example: Prometheus AlertManager (Red Hat OpenShift) generic webhook event mappings

Review the following example mappings for a generic webhook connection between IBM Cloud Pak for AIOps and Prometheus AlertManager (Red Hat OpenShift). Use these mappings to help you configure your own event mapping for a generic webhook connection.

- Example event (alert) JSON
- Example event mapping JSONata expression
- Example event result JSON
- Example Curl Command

Example event (alert) JSON

```
{
  "receiver": "Default",
  "status": "firing",
  "alerts": [
    {
      "status": "firing",
      "labels": {
        "alertname": "PrometheusMissingRuleEvaluations",
        "container": "kube-rbac-proxy",
        "endpoint": "metrics",
        "instance": "10.254.40.166:9092",
        "job": "prometheus-k8s",
        "namespace": "openshift-monitoring",
        "openshift_io_alert_source": "platform",
        "pod": "prometheus-k8s-1",
        "prometheus": "openshift-monitoring/k8s",
        "rule_group": "/etc/prometheus/rules/prometheus-k8s-rulefiles-0/openshift-kube-
apiserver-kube-apiserver-slos-94dbc1dd-b771-4899-8c8f-54c0f5353e04.yaml;kube-apiserver.rules",
        "service": "prometheus-k8s",
        "severity": "warning"
      },
      "annotations": {
        "description": "Prometheus openshift-monitoring/prometheus-k8s-1 has missed 1 rule
```

```
group evaluations in the last 5m.",
        "summary": "Prometheus is missing rule evaluations due to slow rule group evaluation."
      },
      "startsAt": "2023-06-01T17:02:53.691Z",
      "endsAt": "0001-01-01T00:00:00Z",
      "generatorURL": "https://prometheus-k8s-openshift-monitoring.apps.example.ibm.com/graph?
g0.expr=increase%28prometheus_rule_group_iterations_missed_total%7Bjob%3D~%22prometheus-
k8s%7Cprometheus-user-workload%22%7D%5B5m%5D%29+%3E+0&g0.tab=1",
      "fingerprint": "24ad9e973e22bdce"
    },
    {
      "status": "firing",
      "labels": {
        "alertname": "AlertmanagerReceiversNotConfigured",
        "namespace": "openshift-monitoring",
        "openshift_io_alert_source": "platform",
        "prometheus": "openshift-monitoring/k8s",
        "severity": "warning"
      },
      "annotations": {
        "description": "Alerts are not configured to be sent to a notification system, meaning
that you may not be notified in a timely fashion when important failures occur. Check the
OpenShift documentation to learn how to configure notifications with Alertmanager.",
        "summary": "Receivers (notification integrations) are not configured on Alertmanager"
      },
      "startsAt": "2023-04-26T19:27:36.642Z",
      "endsAt": "0001-01-01T00:00:00Z",
      "generatorURL": "https://prometheus-k8s-openshift-monitoring.apps.example.ibm.com/graph?
g0.expr=cluster%3Aalertmanager_integrations%3Amax+%3D%3D+0&g0.tab=1",
      "fingerprint": "72bc0ebbd3167d00"
    }
  ],
  "groupLabels": {
    "namespace": "openshift-monitoring"
  },
  "commonLabels": {
    "namespace": "openshift-monitoring",
    "openshift_io_alert_source": "platform",
    "prometheus": "openshift-monitoring/k8s",
    "severity": "warning"
  },
  "commonAnnotations": {},
  "externalURL": "https:///console-openshift-console.apps.example.ibm.com/monitoring",
  "version": "4",
  "groupKey": "{}:{namespace=\"openshift-monitoring\"}",
  "truncatedAlerts": 0
}
```

Example event mapping JSONata expression

```
(
    /* Set resource based on labels available */
    $resource := function($labels){(
      $name := $labels.name  ? $labels.name
        : $labels.statefulset  ? $labels.statefulset
        : $labels.deployment  ? $labels.deployment
        : $labels.daemonset  ? $labels.daemonset
        : $labels.pod  ? $labels.pod
        : $labels.container  ? $labels.container
        : $labels.instance  ? $labels.instance
        : $labels.app  ? $labels.app
        : $labels.job_name  ? $labels.job_name
        : $labels.job  ? $labels.job
        : $labels.type ? $labels.type: $labels.prometheus;
        $labels.namespace ? ($name & '/' & $labels.namespace): $name;
      )
    };
    /* Map to event schema */
    alerts.(
      {
        "summary": annotations.summary ? annotations.summary: annotations.description ?
```

```
annotations.description : annotations.message ? annotations.message,
        "severity": $lowercase(labels.severity) = "critical" ? 6 : $lowercase(labels.severity)
= "major" ? 5 : $lowercase(labels.severity) = "minor" ? 4 : $lowercase(labels.severity) =
"warning" ? 3 : 1,
        "resource": {
          "name": $resource(labels)
        },
        "type": {
          "eventType": $lowercase(status) = "firing" ? "problem": "resolution",
          "classification": labels.alertname
        },
        "links": [
          {
              "url": generatorURL
          }
        ],
        "sender": {
          "name": "Prometheus",
          "type": "Webhook Connector"
        },
       "details": labels
      }
    )
  )
```

Example event result JSON

The following example shows the JSON result that creates an array of events. The generic webhook connector creates a CloudEvent for each element in the array.

```
[
    {
        "occurrenceTime": "2023-06-07T14:49:53Z",
        "summary": "Prometheus is missing rule evaluations due to slow rule group
evaluation.",
        "severity": 3,
        "type": {
            "eventType": "problem",
            "classification": "PrometheusMissingRuleEvaluations",
            "condition": null
        },
        "sender": {
            "name": "Prometheus",
            "type": "Webhook Connector"
        },
        "resource": {
            "name": "prometheus-k8s-1/openshift-monitoring"
        },
        "links": [
            {
                "linkType": null,
                "name": null,
                "description": null,
                "url": "https://prometheus-k8s-openshift-
monitoring.apps.example.ibm.com/graph?
g0.expr=increase%28prometheus_rule_group_iterations_missed_total%7Bjob%3D~%22prometheus-
k8s%7Cprometheus-user-workload%22%7D%5B5m%5D%29+%3E+0&g0.tab=1"
            }
        ],
        "details": {
            "alertname": "PrometheusMissingRuleEvaluations",
            "container": "kube-rbac-proxy",
            "endpoint": "metrics",
            "instance": "10.254.40.166:9092",
            "job": "prometheus-k8s",
            "namespace": "openshift-monitoring",
            "openshift_io_alert_source": "platform",
            "pod": "prometheus-k8s-1",
            "prometheus": "openshift-monitoring/k8s",
            "rule_group": "/etc/prometheus/rules/prometheus-k8s-rulefiles-0/openshift-kube-
apiserver-kube-apiserver-slos-94dbc1dd-b771-4899-8c8f-54c0f5353e04.yaml;kube-apiserver.rules",
```

```
            "service": "prometheus-k8s",
            "severity": "warning"
        }
    },
    {
        "occurrenceTime": "2023-06-07T14:49:53Z",
        "summary": "Receivers (notification integrations) are not configured on Alertmanager",
        "severity": 3,
        "type": {
            "eventType": "problem",
            "classification": "AlertmanagerReceiversNotConfigured",
            "condition": null
        },
        "sender": {
            "name": "Prometheus",
            "type": "Webhook Connector"
        },
        "resource": {
            "name": "openshift-monitoring/k8s/openshift-monitoring"
        },
        "links": [
            {
                "linkType": null,
                "name": null,
                "description": null,
                "url": "https://prometheus-k8s-openshift-
monitoring.apps.example.ibm.com/graph?
g0.expr=cluster%3Aalertmanager_integrations%3Amax+%3D%3D+0&g0.tab=1"
            }
        ],
        "details": {
            "alertname": "AlertmanagerReceiversNotConfigured",
            "namespace": "openshift-monitoring",
            "openshift_io_alert_source": "platform",
            "prometheus": "openshift-monitoring/k8s",
            "severity": "warning"
        }
    }
]
```

Example Curl Command

```
# Webhook details
WEBHOOK_URL=https://webhook-connector-cc1c414f-f356-4ae8-bd61-a85e56d18cf4-
aiops.apps.example.ibm.com/webhook-connector/asy2gjgcebu
AUTH=test:test

# Send payload
curl -X POST -u ${AUTH} --insecure -d '{
  "receiver": "Default",
  "status": "firing",
  "alerts": [
    {
      "status": "firing",
      "labels": {
        "alertname": "PrometheusMissingRuleEvaluations",
        "container": "kube-rbac-proxy",
        "endpoint": "metrics",
        "instance": "10.254.40.166:9092",
        "job": "prometheus-k8s",
        "namespace": "openshift-monitoring",
        "openshift_io_alert_source": "platform",
        "pod": "prometheus-k8s-1",
        "prometheus": "openshift-monitoring/k8s",
        "rule_group": "/etc/prometheus/rules/prometheus-k8s-rulefiles-0/openshift-kube-
apiserver-kube-apiserver-slos-94dbc1dd-b771-4899-8c8f-54c0f5353e04.yaml;kube-apiserver.rules",
        "service": "prometheus-k8s",
        "severity": "warning"
      },
      "annotations": {
        "description": "Prometheus openshift-monitoring/prometheus-k8s-1 has missed 1 rule
```

```
group evaluations in the last 5m.",
        "summary": "Prometheus is missing rule evaluations due to slow rule group evaluation."
      },
      "startsAt": "2023-06-01T17:02:53.691Z",
      "endsAt": "0001-01-01T00:00:00Z",
      "generatorURL": "https://prometheus-k8s-openshift-monitoring.apps.example.ibm.com/graph?
g0.expr=increase%28prometheus_rule_group_iterations_missed_total%7Bjob%3D~%22prometheus-
k8s%7Cprometheus-user-workload%22%7D%5B5m%5D%29+%3E+0&g0.tab=1",
      "fingerprint": "24ad9e973e22bdce"
    },
    {
      "status": "firing",
      "labels": {
        "alertname": "AlertmanagerReceiversNotConfigured",
        "namespace": "openshift-monitoring",
        "openshift_io_alert_source": "platform",
        "prometheus": "openshift-monitoring/k8s",
        "severity": "warning"
      },
      "annotations": {
        "description": "Alerts are not configured to be sent to a notification system, meaning
that you may not be notified in a timely fashion when important failures occur. Check the
OpenShift documentation to learn how to configure notifications with Alertmanager.",
        "summary": "Receivers (notification integrations) are not configured on Alertmanager"
      },
      "startsAt": "2023-04-26T19:27:36.642Z",
      "endsAt": "0001-01-01T00:00:00Z",
      "generatorURL": "https://prometheus-k8s-openshift-monitoring.apps.example.ibm.com/graph?
g0.expr=cluster%3Aalertmanager_integrations%3Amax+%3D%3D+0&g0.tab=1",
      "fingerprint": "72bc0ebbd3167d00"
    }
  ],
  "groupLabels": {
    "namespace": "openshift-monitoring"
  },
  "commonLabels": {
    "namespace": "openshift-monitoring",
    "openshift_io_alert_source": "platform",
    "prometheus": "openshift-monitoring/k8s",
    "severity": "warning"
  },
  "commonAnnotations": {},
  "externalURL": "https:///console-openshift-console.apps.example.ibm.com/monitoring",
  "version": "4",
  "groupKey": "{}:{namespace=\"openshift-monitoring\"}",
  "truncatedAlerts": 0
}' -H 'Content-Type: application/json' ${WEBHOOK_URL}
```

Creating IBM Tivoli Netcool/Impact connections

You can integrate your existing IBM Tivoli Netcool/Impact policies with IBM Cloud Pak for AIOps through a IBM Tivoli Netcool/Impact connection. This connection connects the IBM Tivoli Netcool/Impact cluster with IBM Cloud Pak for AIOps.

- About this connection
- Before you begin
- Creating IBM Tivoli Netcool/Impact connections
- Editing IBM Tivoli Netcool/Impact connections
- Deleting IBM Tivoli Netcool/Impact connections

About this connection

By connecting with IBM Tivoli Netcool/Impact, IBM Cloud Pak for AIOps can then leverage your existing IBM Tivoli Netcool/Impact deployment to run automation policies that you manage with IBM Tivoli Netcool/Impact. New policies that you create in IBM Tivoli Netcool/Impact can reference the extension framework API for IBM Cloud Pak for AIOps. With these

policies, you can write automation processes in IBM Tivoli Netcool/Impact to complete operations in IBM Cloud Pak for AIOps, such as to query alerts and incidents, and request changes to alerts.

Note: You can create multiple connections between IBM Cloud Pak for AIOps and IBM Tivoli Netcool/Impact. Each connection must point to a unique Impact endpoint.

When you create the connection in IBM Cloud Pak for AIOps to IBM Tivoli Netcool/Impact, the connection automatically creates the RESTful DSA data source in IBM Tivoli Netcool/Impact. To complete the integration with IBM Tivoli Netcool/Impact connection after creating the connection in IBM Cloud Pak for AIOps, you need to edit this RESTful DSA data source in IBM Tivoli Netcool/Impact to define your authentication settings for communicating with IBM Cloud Pak for AIOps API. After editing and verifying the authentication settings, you can begin to create and run your automation policies from IBM Tivoli Netcool/Impact.

Note: If the IBM Tivoli Netcool/Impact cluster uses a non-default cluster name ("NCICLUSTER"), the connector might fail to validate the connection. For more information, see Impact connector fails for Impact server with non-default cluster name.

Before you begin

- Ensure that you have IBM Tivoli Netcool/Impact Version 7.1 or later and IBM Cloud Pak for AIOps 4.2.0 installed.
- Obtain the following details for your IBM Tivoli Netcool/Impact deployment:
  - The endpoint for the GUI server (example: https://noi-impact.fyre.ibm.com:16311)
  - The endpoint(s) for the primary Impact server and any secondary servers (example: https://noi-impact.fyre.ibm.com:9081)
  - If Impact is deployed with Event Manager on OpenShift, you can use the **`oc describe noi`** command to retrieve the endpoints.
  - The username and password for the IBM Tivoli Netcool/Impact administrator (example: impactadmin) for connecting to the IBM Tivoli Netcool/Impact endpoint.
  - The set of CA certificates for IBM Tivoli Netcool/Impact if the endpoint is secured by SSL/TLS. If self-signed this means all the certificates from all the servers.

    You can extract these certificates from your IBM Tivoli Netcool/Impact deployment with **`openssl`**:

    - For the GUI server (port 16311)

      ```
      openssl s_client -showcerts -servername noi-impact.mycluster.com -connect noi-impact.mycluster.com:16311 </dev/null
      ```

    - For the backend server (port 9081)

      ```
      openssl s_client -showcerts -servername noi-impact.mycluster.com -connect noi-impact.mycluster.com:9081 </dev/null
      ```

    To combine multiple certificates (there is a minimum requirement of two) in the connection settings, you can list them sequentially in no specific order. The combined certificates resembles the following structure:

    ```
    -----BEGIN CERTIFICATE-----
    ABCDEfGHIjklMnOPQrSTUVWXYZABCdefghiJ1k2LMNoPQRSTUVwxYZABCDEFGhij
    ... Server 1 ...
    r+wGXNPwrw==
    -----END CERTIFICATE-----
    -----BEGIN CERTIFICATE-----
    ABCDEfGHIjklMnOPQrSTUVWXYZABCdefghiJ1k2LMNoPQRSTUVwxYZABCDEFGhij
    ... Server 2 ...
    ZYXWVUTSRqpoNnLKJiHGFEDCBAZYXwvutsrQ1p4ONMlKJIHGFEdcBAZYXWVUTsrq
    -----END CERTIFICATE-----
    ```

Creating an IBM Tivoli Netcool/Impact connection

To create an IBM Tivoli Netcool/Impact connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.
4. From the list of available connections, find and click the **IBM Tivoli Netcool/Impact** tile.

   Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section (Events).
5. On the side-panel, review the instructions and when ready to continue, click **Connect**.
6. On the *Add connection* page, define the general connection details:

   - **Name**: The display name of your connection. You cannot use spaces or special characters in the display name. Use only alphanumeric characters.
   - **GUI Server URLs of Impact endpoint**: Enter the GUI URL or URLs for the endpoint. For high availability environments with multiple URLs, enter the URLs as a comma-separated list. For example, **`https://impact-1, https://impact-2`**.
   - **Backend Server URLs of Impact endpoint**: Enter the backend URL or URLs for the endpoint. For high availability environments with multiple URLs, enter the URLs as a comma-separated list. For example, **`https://impact-1, https://impact-2`**.
   - **User ID**: Enter the user ID for the IBM Tivoli Netcool/Impact administrator (example: impactadmin) for connecting to the IBM Tivoli Netcool/Impact endpoint.
   - **Password**: Enter the password for the IBM Tivoli Netcool/Impact administrator for connecting to the IBM Tivoli Netcool/Impact endpoint.
   - **Impact Certificates**: Enter the valid set of CA certificates for IBM Tivoli Netcool/Impact to use for verifying the SSL/TLS connection to the REST services for IBM Tivoli Netcool/Impact.
   - **Deployment options**: Select one of the options, **Local** or **Remote**. **Local** deploys the connection in the same cluster and project (namespace) where IBM Cloud Pak for AIOps is installed. Alternatively, **Remote** deploys the connection in your chosen location, such as a different network region, on SaaS, or a remote on-premises VM, cluster, or container.

     Note: If you select **Remote** after adding the connection, you are provided with a bootstrap script to finish the deployment. Download and run the bootstrap script on the remote cluster to complete the deployment.
7. Click **Done**. You are redirected to the *IBM Tivoli Netcool/Impact* page for the connection.

   Your new connection is listed in the table on the page. The connection has an intial status of *Initializing*. After a few minutes, the status changes to *Running* if your connection is successful. With your connection successful and running, you now need to configure the connection in IBM Tivoli Netcool/Impact.
8. Configure the connection in IBM Tivoli Netcool/Impact.

   To complete the connection, you need to edit the RESTful DSA data source in IBM Tivoli Netcool/Impact.

Editing an IBM Tivoli Netcool/Impact connection

After you create your connection, you can edit the connection. For example, if you want to add or change the listed GUI or backend URLs for the Impact endpoint, or to update the User ID or password. To edit a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **IBM Tivoli Netcool/Impact** connection type on the *Manage connections* tab of the *Data and tool connections* page.

4. On the *IBM Tivoli Netcool/Impact* page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.
5. Edit your connection as required. Click **Save** when you are done editing.

Your connection is now edited.

Deleting an IBM Tivoli Netcool/Impact connection

If you no longer need your connection and want to delete it entirely, you can delete the connection from the console.

To delete a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **IBM Tivoli Netcool/Impact** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the *IBM Tivoli Netcool/Impact* connections page, click the *Delete* icon for the connection that you want to delete.
5. Enter the name of the connection to confirm that you want to delete your connection. Then, click **Delete**.

   Your connection is deleted in IBM Cloud Pak for AIOps. To fully remove the connection, log into the IBM Tivoli Netcool/Impact GUI and delete the IBM Cloud Pak for AIOps data source and policies from IBM Tivoli Netcool/Impact.

Configuring a connection in IBM Tivoli Netcool/Impact for IBM Cloud Pak for AIOps

Edit the RESTful DSA data source in IBM Tivoli Netcool/Impact for a connection to IBM Cloud Pak for AIOps to finish configuring the connection and integration between IBM Tivoli Netcool/Impact and IBM Cloud Pak for AIOps.

- Before you begin
- Configuring IBM Tivoli Netcool/Impact RESTful DSA data source
- Verifying the connection

Before you begin

1. Ensure that you have created the IBM Tivoli Netcool/Impact connection in the IBM Cloud Pak for AIOps UI. The connection can be created from the Data and tool connections page in the UI. For more information, see Creating IBM Tivoli Netcool/Impact connections.
2. Obtain the following details for your IBM Cloud Pak for AIOps deployment:
   - Your IBM Cloud Pak for AIOps hostname (cluster_address). For more information about obtaining this information, see Accessing your cluster by using the console.
   - Your encoded (base64 format) Platform UI API Key (**`ZenApiKey`**) for accessing IBM Cloud Pak for AIOps APIs. If you do not have an existing API key and need to generate a new API key, complete the following steps:

     Accessing APIs - Get authentication by using Platform UI API Key

     You can verify that your API key works by running the following command:

     ```
     curl -kv -H "Authorization: ZenApiKey ${ZenApiKey}" -H "x-tenant-id: ${tenant_id}"
      "https://${CPD_ROUTE}/aiops/api/issue-resolution/v1/alerts/"
     ```

     Important: If you lose your API key, repeat the preceding steps to generate a new API key. Your old API key becomes invalid, and any applications or scripts cannot authenticate to the platform until you provide your new

API key. For more information, see Accessing APIs.

- The Red Hat OpenShift Container Platform certificate for IBM Cloud Pak for AIOps.

Note: When you create an IBM Tivoli Netcool/Impact connection, the connector creates a Data Source name (DSA) in the IBM Tivoli Netcool/Impact Data Model (as before). In 4.2.0, the DSA name is created in the format of ***`AIOPs-ConnectorName-ConnectorID`***. Any IBM Tivoli Netcool/Impact policies will now use the new DSA instead of the old one. Therefore, you must update the new DSA with your cluster **Host Name** and **Authorization** (see step 4 below).

Configuring the RESTful DSA data source

1. Open the IBM Tivoli Netcool/Impact GUI.
2. Click and view the **Policies** tab.

   If the connection in IBM Cloud Pak for AIOps is successful, a RESTful DSA data source is already created. In addition, the following default policies are already created.

   - **`AIOPS_HandleAction`** - This file defines the wrapper that is used when IBM Cloud Pak for AIOps invokes an action in IBM Tivoli Netcool/Impact.
   - **`AIOPS_Mappings`** - This file defines how events are sent, and mapped, to IBM Cloud Pak for AIOps.
   - **`AIOPS_Utils`** - This file lists the APIs that are used for communicating with IBM Cloud Pak for AIOps.

   Ensure that these policies exist before proceeding. If they are missing, re-create the connection in IBM Cloud Pak for AIOps.
3. Click the **Data Model** tab.

   An **`AIOps-<identifier>`** RESTful DSA data source is now listed. Click this data source to view the Restful API Data Source Editor for the data model.
4. Go to the **Source** section and update the **Host Name** field with the name of the CPD route value.
5. Update the Request Header authorization setting.
   1. Go to the **Protected Request Headers** section.
   2. Select the **Authorization** header and click **Edit** in the table header.
   3. Update the **Header Value** field to include your encoded Platform UI API Key (**`ZenApiKey`**). This value must be in the following format:

      ```
      ZenApiKey <ZenApiKey>
      ```

      Where **`<ZenApiKey>`** is your encoded Platform UI API Key.
   4. Click **OK**.
6. Import the Red Hat OpenShift Container Platform certificate for IBM Cloud Pak for AIOps.

   To get the certficate, use the following steps:

   1. Export certificate from AIOps api host:

```
openssl s_client -showcerts -servername <cpd_route_name> -connect <cpd_route_name>:443
</dev/null
```

   This returns the certificate in the display. The certificate has the following format:

```
-----BEGIN CERTIFICATE-----
ABCDEfGHIjklMnOPQrSTUVWXYZABCdefghiJ1k2LMNoPQRSTUVwxYZABCDEFGhij
... certificate text...
... Server 1 ...
r+wGXNPwrw==
-----END CERTIFICATE-----
```

2. Copy this text into a cert file and import the file to the Impact server:

```
$IMPACT_HOME/sdk/bin/keytool -import -v -keystore
$IMPACT_HOME/wlp/usr/servers/ImpactUI/resources/security/trust.jks -storepass impactadmin
-alias aiops -file <exported_aiops_certificate>
```

7. Click the **Save** icon to save your RESTful DSA data source configuration. Then, test your connection.

Verifying the connection

1. Create a test event policy in IBM Tivoli Netcool/Impact to create an alert in IBM Cloud Pak for AIOps.

   Your IBM Tivoli Netcool/Impact policies must be written in JavaScript to work with IBM Cloud Pak for AIOps. The Impact Policy Language (IPL) is not supported through the connection.

   You can use the following example JavaScript structure to help you create your own test policy. Replace the variables **<variable>** with the values for your own test event and system details.

```
Load('AIOPS_Utils');
occurrenceTime = new Date();
var myEventInstance = "<instance_name>";
var event = {
    deduplicationKey: myEventInstance + " <event_name>",
    sender: {
      service: "<service>",
      name: "<sender_name>",
      type: "Success",
      location: "<location>",
      cluster: "\"quotes\"",
      controller: "<script>alert(\"<sender_controller>\");</script>"
    },
    resource: {
      name: myEventInstance + "<resource_name>",
      type: myEventInstance + " <type>",
      service: myEventInstance + " <service>",
      location: "<location>",
      controller: "<resource_controller>",
      interface: "<interface>",
      ipAddress: "<ip_address>",
      hostname: "<hostname>",
      cluster: "<cluster>"
    },
    type: {
      classification: myEventInstance + " <classification>",
      eventType: "<event_type>"
    },
    severity: 6,
    summary: myEventInstance + " <summary>",
    occurrenceTime: occurrenceTime.toISOString()
  };
var res = aiopsUtils.postEventDSA( "AIOps-<UUID>" , event );
Log("res: " + res);
```

   To obtain the **UUID** value to use for the **aiopsUtils.postEventDSA("AIOps-<UUID>" , event)** setting, run the following command:

```
oc get connectorconfiguration -o json | jq -r '.items[] | (.metadata.name + " - " +
.metadata.uid)'
```

   Note: You need **jq** to run this command. To install **jq**, see Download jq.

   Output:

```
impact-netcool - 123abcd4-1ab2-1a23-1a2b-1aabbc234de5
```

   The value **123abcd4-1ab2-1a23-1a2b-1aabbc234de5** is the UUID value.

2. Run the policy to test the connection from the IBM Tivoli Netcool/Impact UI **Policies** tab.

Open the test policy that you created, and click the **Run** icon in the toolbar. Wait a few minutes, then proceeed to the next step.

3. Open the IBM Cloud Pak for AIOps UI and from the main navigation click **Incidents and alerts**. Click the **Alerts** tab to open the Alert Viewer.

   The alert for your test event should now display.

What to do next

- Create policies in IBM Tivoli Netcool/Impact. For more information about setting up these policies, see the IBM Tivoli Netcool/Impact documentation.
- Create a policy in IBM Cloud Pak for AIOps and invoke an IBM Tivoli Netcool/Impact policy. For more information, see:
  - Creating policies
  - Invoke an Impact policy

Creating Instana connections

You can integrate with Instana to enable connectivity to Instana events, metrics, and topology by using an Instana connection. Instana assets can be combined with IBM Cloud Pak for AIOps AI, such as runbooks, metric anomaly detection, and log anomaly detection to help you gain insights.

If needed, you can create multiple Instana connections. This deployment model has only one topology processor and multiple Instana connectors. The topology processor treats each connector as a distinct topology provider.

The list of technologies is determined dynamically, based on the actual resources available in the Instana instance the connector is pulling metrics data from. Each technology includes a list of all available metrics that can be dynamically retrieved by the Instana connector. The connector also provides a set of recommended metrics to select from.

For more information about working with Instana connections, see the following sections:

- Before you begin
- Creating Instana connections
- Enabling Instana connections
- Editing Instana connections
- Deleting Instana connections

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Before you begin

Ensure that you have following information to hand:

- Endpoint
- API Token
- Filters to define subsets of data that is pulled from Instana (optional)

Creating Instana connections

To create an Instana connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.

4. From the list of available connections, find and click the **Instana** tile.

   Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section.

5. On the side-panel, review the instructions and when ready to continue, click **Connect**.

6. On the *Add connection* page, define the general connection details:

   - **Name**: The display name of your connection.
   - **Description**: An optional description for the connection.
   - **Endpoint**: Enter an endpoint for the connection.
   - **API token**: Enter an API token for the connection. The API Token can be fetched from the Instana web UI.
   - **Deployment options**: Select between the *local* or *remote* radio buttons. If you select remote, after adding the connection, you will be provided with a bootstrap command to finish the deployment.

     Note: To use the Instana API for the collection of Instana data (metrics, events, and topologies), an API token is required. These API tokens can be created on the Instana user interface. Click the **Settings** icon, then **Team Settings > API Tokens**, and lastly click **Add API Token** to create a new API token.

   The default configuration for a new API token is enough for this Instana Connector.

7. Click **Next**.

8. If you would you like to optionally use an HTTP proxy configuration with the Instana connection, enter the following *Configure proxy* information:

   - **Proxy URL**: Enter the URL or domain name of the Proxy Server including the protocol. Do not include the port in the Proxy URL field.
   - **Proxy Port**: Enter the port as a number in the Proxy Port field.
   - **Proxy Username**: If the HTTP proxy server requires authentication, enter the proxy username. If not, you can leave this field blank to use the proxy without authentication.
   - **Proxy Password**: If the HTTP proxy server requires authentication, enter the proxy password. If not, you can leave this field blank to use the proxy without authentication.

   Note: An HTTPS proxy server and certificate authentication are not supported for the proxy configuration. Only basic authentication (username and password) or no authentication with an HTTP proxy server are supported.

9. Click **Next**.

10. Enter the following *Collect topology data* information:

- **Enable data collection**: Enable the toggle button.
- **Time window (seconds)**: Reference the windowSize within the Instana API.
- **Connection Intervals (seconds)**: Specify a time period for how frequently to run the job.
- **Application name allowlist pattern**: Filter app perspective collection based on the name of the application.
- **Import Instana application perspectives as Cloud Pak for AIOps applications**: Enable the toggle button (if required).

  Note: When this toggle button is enabled, an IBM Cloud Pak for AIOps application for each application perspective in Instana is automatically created. If it is not turned on, then you create resource groups for only the services that are part of that application in Instana. So, you have the opportunity to define your own IBM Cloud Pak for AIOps applications, by adding your own group to the application. It's important to note that if the application is created automatically by enabling the toggle button, the application is not controlled by the user. It is the product itself that controls the application. Only if you create the application yourself can you control the lifecycle of the application, and thus remove or delete it.

11. Click **Next**.

12. Enter the following *Collect event data* information:

- **Enable data collection**: Enable the toggle button if required.
- **Types of events**: Select one or more event types for data collection from the following options:
    - Incidents
    - Issues
    - Changes

13. Click **Next**.
14. Enter the following *Collect metric data* information to define how you want to collect metric data. Topology metric data that is related to applications, endpoints, and services can be enabled and thus collected, or indeed disabled.

- **Enable data collection**: Set the toggle button to 'on'.
- **Monitored metrics**: You can select built-in metrics to monitor. These metrics are categorized by the technologies that they are associated with. You can also add custom-defined metrics to monitor. To select metrics, use any of the following options:
    - Enter your preferred metrics in the search bar and select their corresponding checkboxes.
    - Search and select your preferred technologies and their associated metrics in their respective filter table. To find the related metrics of a technology, click its corresponding arrow. By default, all metrics of the chosen technologies are selected.
    - Select the **Start from recommended metrics** checkbox if you are not sure which metrics to start with. This option selects all recommended technologies and their associated metrics in the filter table.

  Note:
    - Clear the selected checkboxes to undo any selection. However, if you clear the **Start from recommended metrics** checkbox, all current selection is cleared.
    - All selected metrics are listed in the **Selected metrics** list. You can remove any listed metric by clicking the **Deselect Metric** icon corresponding to it.
    - You might see performance issues or gaps in your data due to the number of metrics selected. To avoid this problem, consider starting with the set of recommended metrics or those that are most important for you to monitor.

15. Enter the following *Metric data summary* information:

    - **Selected metrics**: The table lists all the selected technologies and metrics. You can remove any metrics from the list. To remove a listed metric, select the checkbox corresponding to it and click **Remove**.
    - **Only collect Live Data once the connection is established**: Select the checkbox to enable live data collection. **Note**: If you choose to collect only live data, you cannot collect historical data after the live data collection starts.
    - **Aggregation interval (minutes)**: Enter a value in minutes. This value is the frequency at which data is collected.
    - **Historical data collection**: The recommended time period for collecting historical metric data is two weeks, though it is possible to collect it for up to 31 days. Enter a start and end date and time in the following formats:
        - *Start date and time**: Select the start date from the calendar and enter the time in the 'hh:mm' format (hours and minutes).

          Note: Your start date should not exceed 31 days from the present.
        - **Time zone**: Select your time zone from the dropdown list.
        - **End date and time**: Click **Add end date** and select a date from the calendar. Also, enter the time in the 'hh:mm' format (hours and minutes). If you don't want to set an end date at this stage, click **Remove End Date**.
        - **Aggregation interval (minutes)**: Enter a value in minutes. This value is the frequency at which data is collected.

      Note: If you do not select an end date, IBM Cloud Pak for AIOps collects historical data from the specified start date and continues with live data collection. You cannot collect historical data after live data collection starts.

Note: If you have already set an end date and time for an Instana connection, you cannot edit it. If you need to edit either of these properties, you must delete your connection and create a new one.

16. Click **Done**.

For more information about performance considerations for metric data collection, see Performance considerations for metric data collection.

You have created an *Instana* connection in your instance.

After the connection creation is complete, and if you selected the *Remote* option for connector deployment, you can download the *bootstrap script*. Select the created connection, and then click **download** from the options menu (three vertical dots). If a script is regenerated, it invalidates the credentials of previously downloaded scripts for that connection.

Note: You can either copy the script or download as an sh file to run it on a remote cluster.

Prerequisites before you run the bootstrap script on a remote cluster

- Podman needs to be installed on the remote cluster.
- Log in to the *Image registry* for the script to pull in the image by using the Podman login command.

After you create your connection, you must enable the data collection to connect your connection with the AI of IBM Cloud Pak for AIOps. For more information about enabling your connection, see Enabling Instana connections.

To create more connections (such as a ChatOps connection), see Configuring data and tool connections.

For more information about working with the insights provided by your connections, see ChatOps insight management.

For Instana on-prem installation, the recommended rate limit is double the number of resources. This limit is enough to cover most cases, although situations might arise where the limits need to be increased even more. When a more precise way to determine the limit is required (for instance a shared multi-tenant SaaS environment), the following formula can be applied:

```
number-of-metric-API-calls-per-hour ~= (number-of-selected-technologies x 2) x (snapshots-for-selected-technologies / 30) x (60 / collection-interval)
number-of-topology-API-calls-per-hour ~= (number-of-application-perspectives x (60 / collection-interval)) +(number-of-services x (60 / collection-interval))
number-of-events-API-calls-per-hour = 60

total= number-of-metric-API-calls-per-hour + number-of-topology-API-calls-per-hour + number-of-events-API-calls-per-hour
```

Note: This formula is not always 100% accurate, as each technology or plug-in has a different number of metrics collected (the mean value across these is used, which is 2 collection cycles or technology). If the environment is unbalanced toward either end of the spectrum (for instance, if you have mostly hosts that define the most metrics), then the formula might underestimate the required limit.

To determine the number of resources (snapshots) for each infrastructure pluginAPI, you can use the following:

```
api/infrastructure-monitoring/snapshots?plugin=technology_name
```

Example:

```
api/infrastructure-monitoring/snapshots?plugin=host
```

For more information about Instana APIs, see Instana API.

With the following example curl commands, you can retrieve the number of the following resources:

- **`snapshots-for-selected-technologies`** (such as host)

  ```
  curl -k -s --request GET 'https://<instana server hostname>/api/infrastructure-monitoring/snapshots?plugin=host' --header 'Authorization: apiToken <api token>' | jq '.items|length'
  ```

- **`number-of-application-perspectives`**

  ```
  curl -k -s --request GET 'https://<instana server hostname>/api/application-monitoring/applications' --header 'Authorization: apiToken <api token>' | jq '.items|length'
  ```

- `number-of-services`

  ```
  curl -k -s --request GET 'https://<instana server hostname>/api/application-monitoring/services' --header 'Authorization: apiToken <api token>' | jq '.items|length'
  ```

Enabling and disabling Instana connections

If you didn't enable your data collection during creation, you can enable your connection afterward. You can also disable a previously enabled connection the same way. To enable or disable a created connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Manage connections* tab of the *Data and tool connections* page, click the **Instana** connection type.
4. Click the connection that you want to enable or disable.
5. Go to the *Collect topology data* section. Enable the toggle button if required.
6. Go to the *Collect event data* section. Enable the toggle button if required.
7. Go to the *Collect metric data* section. Enable the toggle button if required.

Note: Disabling data collection for a connection does not delete the connection.

You have enabled or disabled your connection. For more information about deleting a connection, see Deleting Instana connections.

Editing Instana connections

After you create your connection, you can edit it. To edit a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Instana** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the Instana connections page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.
5. Edit your connection as required. Click **Save** when you are done editing.

You have edited your connection. If you have not previously enabled or disabled your application, you can enable or disable the connection directly from the interface. For more information about enabling and disabling your connection, see Enabling and disabling Instana connections. For more information about deleting a connection, see Deleting Instana connections.

Deleting Instana connections

If you no longer need your Instana connection and want to not only disable it, but delete it entirely, you can delete the connection from the console.

To delete a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Instana** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the Instana connections page, click the options menu (three vertical dots) for the connection that you want to delete and click **Delete**.

5. Enter the name of the connection to confirm that you want to delete your connection. Then, click **Delete**.

Your connection is deleted.

Creating Kafka connections

A Kafka connection can provide either event or log data. Log data can be used to establish a baseline of normal behavior and then identify anomalies. Event data enables the analysis and processing of different types of alerts and events. Anomalies that are identified can be correlated with alerts and events and published to your ChatOps interface to help you determine the cause and resolution of a problem.

Custom connections can route only a single source of data at one time. Unlike the custom connection type, you can use the Kafka connection type to collect log data and event data from different systems. Then, you can route that information through a forwarding agent, such as an Apache Kafka topic like Sysdig or FluentBit.

You can also use the Kafka connection to enable training with an offline, historical data set. For more information about using Kafka topics for offline training, see Importing offline log data.

Note: The Kafka replication factor is set to one replica by default. If you are implementing a production deployment of IBM Cloud Pak for AIOps, you might lose data if your Kafka pods fail or restart. If the data collection is enabled in your Kafka connection when the Kafka pods go down, you might experience a gap in the data that your connection generated during that down period.

For more information about working with Kafka connections, see the following sections:

- Creating Kafka connections
- Enabling Kafka connections
- Editing Kafka connections
- Importing offline log data
- Deleting Kafka connections

Creating Kafka connections

Unlike other connection types, you can use Kafka connections to push properly formatted historical training data to IBM Cloud Pak for AIOps. To create a Kafka connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.
4. From the list of available connections, find and click the **Kafka** tile.

   Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Connect**.
6. On the *Add connection* page, define the general connection details:
   - **Name**: The display name of your Kafka connection.
   - **Description**: An optional description for the Kafka connection.
   - **Kafka partitions**: The number of Kafka partitions. The default value of `1` might be suitable for importing a few log records. However, if you intend to import many records, increase this number. The range is `1` to `500`. For example, for a proof of content (PoC) deployment where you need to import a large data set, you can use `48` Kafka partitions.
   - **JSON processing option**: Select a JSON processing option.

- **None**: The default option. The JSON is not processed or modified.
- **Flatten**: This option flattens the JSON object by removing the opening and closing braces.
- **Filter**: This option extracts the JSON object and replaces it with an empty string.
- For more information about the options, see Managing embedded JSON.

7. Click **Next**.

Entering Field mapping information

Unlike other connection types, you can specify what kind of information you want to run through your Kafka connection. For example, if you want to collect data from multiple custom sources, you can set them up as separate Kafka connections.

1. In the **Field mapping** section, set the **Data source** to specify the type of data incoming from the Kafka connection. You can select **Events** or **Logs**.
2. Specify the **Mapping type**, which creates the field mapping for your specified log type.
   - Mapping types for **Events** include **None**, **PagerDuty**, and **NOI**.
   - Mapping types for **Logs** are **None**, **ELK**, **Falcon LogScale**, **LogDNA**, **Custom**, and **Splunk**.
     - If you have log data that is already normalized, choose **None**.
     - If your log data is in one of the supported log formats, such as ELK or Falcon LogScale, choose the corresponding log format.
     - For field mapping for **Custom** types, see Creating custom connections.
3. The **Topic** name for a Kafka connection is predefined. Note this topic and send your data to it.
4. For log connections, complete the following settings:
   1. Enter the **Maximum number of logs per second** rounded to the nearest thousand. The maximum number of logs per second is collected in increments of 1,000 up to 25,000 and is rounded up to the nearest thousand sent from Kafka to IBM Cloud Pak for AIOps.
   2. Set the **Field mapping** mode to **Live data for initial AI training** for initial training or to **Live data for continuous AI training and anomaly detection**.
      - You can improve search performance by mapping the fields from your implementation fields to standard IBM Cloud Pak for AIOps fields. For more information about how field mappings are defined, see Mapping data from incoming sources.
      - For more information about using mappings to clean your data for use in IBM Cloud Pak for AIOps, see Cleaning mapped data using regular expressions.
   3. In the **Mapping** section, if you use a **Mapping type** other than **None**, verify whether the JSON document that is automatically assigned to this field is valid. The validity depends on the structure of your log records. For example, for a PoC deployment, an example default **Field mapping** for ELK can resemble the following mapping:

```
{
  "codec": "elk",
  "message_field": "@message",
  "log_entity_types": "@hostname, @bundleName, @context.Environment",
  "instance_id_field": "@properties.processtype",
  "rolling_time": 10,
  "timestamp_field": "@timestamp"
}
```

   If you chose **None** for the **Mapping type**, each log record in your log files conforms to the following format. Set both of the `application_id` and `application_group_id` values to `"1000"`, and make sure that a 13-digit timestamp value is assigned to the `timestamp` field:

```
{
    "timestamp": 1581549409000,
    "utc_timestamp": "2020-02-12 23:16:49.000",
    "instance_id": "calico-node-jn2d2",
    "application_group_id": "1000",
    "application_id": "1000",
    "features": [],
    "meta_features": [],
```

```
    "level": 1,
    "message": "[64] ipsets.go 254: Resyncing ipsets with dataplane.
family=\"inet\"",
    "entities": {
        "pod": "calico-node-jn2d2",
        "cluster": null,
        "container": "calico-node",
        "node": "kube-bmgcm5td0stjujtqv8m0-ai4itsimula-wp16cpu-00002c34"
    },
    "type": "StandardLog"
}
```

Important: The **Events** data that is collected must follow the **Kafka Connection Normalized Event schema**. For more information, see Normalized mapping rules.

5. Click **Next**.

Note: When importing events the JSON structure for an event must be on a single line in the file. For more information about the structure for importing event data, see Normalized Mapping Rules.

Entering AI training and log data information

1. If you want to enable the connection or if you want to enable data collection for AI training and log data, switch the **Data collection** toggle to **On**.
2. Click **Save** to save your connection.

You created a Kafka connection in your instance, and you can now use the Kafka connection as the basis for your offline AI model training. After you create your connection, enable the data collection to connect your connection with the AI of IBM Cloud Pak for AIOps.

- For more information about enabling your connection, see Enabling Kafka connections.
- To create more connections, such as a ChatOps connection, see Configuring data and tool connections.
- For more information about working with the insights provided by your connections, see ChatOps insight management.
- For more information about AI model training, see Configuring AI models.

Importing offline log data

Before you proceed with the following steps, make sure that the data flow is enabled for the Kafka Ops integration that you previously defined. You can import offline log data into IBM Cloud Pak for AIOps in two ways. Both mechanisms publish the offline log data to the Kafka topic that is associated with the Kafka Ops integration.

Using a Kafka utility to publish log data

1. Install a Kafka utility, such as the kcat utility.

2. Log in to the Red Hat® OpenShift® cluster where IBM Cloud Pak for AIOps is installed.

3. Run the following commands from the namespace where IBM Cloud Pak for AIOps is installed:

```
oc extract secret/iaf-system-cluster-ca-cert --keys=ca.crt --to=- > ca.crt

export sasl_password=$(oc get secret cp4waiops-cartridge-kafka-auth-0 --template=
{{.data.password}} | base64 --decode);

export BROKER=$(oc get routes iaf-system-kafka-bootstrap -
o=jsonpath='{.status.ingress[0].host}{"\n"}'):443
```

4. Publish the log data to the Kafka topic. This example uses kcat. If you also use kcat, update the variables and run the following commands:

```
export KAFKA_TOPIC=<kafka topic>

export LOG_FILE=<log file>

kcat -X security.protocol=SASL_SSL -X ssl.ca.location=ca.crt -X sasl.mechanisms=SCRAM-
SHA-512 -X sasl.username=cp4waiops-cartridge-kafka-auth-0 -X sasl.password=$sasl_password
-X enable.ssl.certificate.verification=false -b $BROKER -P -t $KAFKA_TOPIC -l $LOG_FILE
```

Copying log data into the Kafka cluster

Use this option only if you can't use a Kafka utility such as kcat for your environment. Complete the following steps to publish the log data to the corresponding Kafka topic by copying the log data into the Kafka cluster:

1. Run the following commands from the namespace where IBM Cloud Pak for AIOps is installed to obtain the Kafka password and Kafka broker. Record both values, which you need in a following step.

   ```
   export kafka_password=$(oc get secret cp4waiops-cartridge-kafka-auth-0 --template=
   {{.data.password}} | base64 --decode)

   export kafka_broker=$(oc get routes iaf-system-kafka-bootstrap -
   o=jsonpath='{.status.ingress[0].host}{"\n"}'):443

   echo $kafka_password

   echo $kafka_broker
   ```

2. Copy the log file to the Kafka cluster:

   ```
   oc cp <log file> iaf-system-kafka-0:/tmp
   ```

3. Check that the file is copied to the Kafka cluster from the namespace where IBM Cloud Pak for AIOps is installed:

   ```
   oc exec -it iaf-system-kafka-0 -- bash

   ls /tmp/<log file>

   exit
   ```

4. Get the `ca.cert` certificate file:

   ```
   oc extract secret/iaf-system-cluster-ca-cert --keys=ca.crt --to=- > ca.crt
   ```

5. Convert `ca.crt` to Java keystore format (JKS):

   ```
   keytool -import -trustcacerts -alias root -file ca.crt -keystore truststore.jks -
   storepass password -noprompt
   ```

6. Copy the truststore file to the Kafka cluster:

   ```
   oc cp truststore.jks iaf-system-kafka-0:/tmp
   ```

7. Give bash access to the Kafka cluster again:

   ```
   oc exec -it iaf-system-kafka-0 -- bash
   ```

8. Create a file within the Kafka pod that is called `producer.properties`. The contents of the `producer.properties` file can resemble the following properties. Update the `variable placeholders` before you use this content. The `ssl.truststore.location` value points to the JSK file that you created earlier.

   ```
   security.protocol=SASL_SSL
   ssl.truststore.location=/tmp/truststore.jks
   ssl.truststore.password=password
   sasl.mechanism=SCRAM-SHA-512
   sasl.jaas.config=org.apache.Kafka.common.security.scram.ScramLoginModule required \
   username="cp4waiops-cartridge-kafka-auth-0" \
   password="<kafka password>"
   ```

9. From the pod, run the Kafka producer script to publish the log data to the corresponding Kafka topic. Update the `variable placeholders`:

   ```
   /opt/Kafka/bin/Kafka-console-producer.sh --broker-list <kafka broker> --producer.config
   producer.properties --topic <kafka topic> < <log file>
   ```

Enabling and disabling Kafka connections

If you didn't enable your data collection during creation, you can enable your connection after. You can also disable a previously enabled connection the same way. If you selected **Live data for initial AI training** when you created your

connection, disable the connection before AI model training. To enable or disable a created connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Manage connections* tab of the *Data and tool connections* page, click the **Kafka** connection type.
4. Click the connection that you want to enable or disable.
5. Go to the *AI training and log data* section. Set **Data connection** to **On** or **Off** to enable or disable data collection. Disabling data collection for a connection does not delete the connection.

You enabled or disabled your connection. For more information about deleting a connection, see Deleting Kafka connections.

Editing Kafka connections

After you create your connection, you can edit the connection. For example, if you want to refine the field mapping for a Kafka connection from a Falcon LogScale source, you can edit it. To edit a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Kafka** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the Kafka connections page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.
5. Edit your connection as required. Click **Save** when you are done editing.

Your connection is now edited. If your application was not previously enabled or disabled, you can enable or disable the connection directly from the interface. For more information about enabling and disabling your connection, see Enabling and disabling Kafka connections. For more information about deleting a connection, see Deleting Kafka connections.

Deleting Kafka connections

If you no longer need your Kafka connection and want to not only disable it, but delete it entirely, you can delete the connection from the console.

Note: You must disable data collection before you delete your connection. For more information about disabling data collection, see Enabling and disabling Kafka connections.

To delete a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Kafka** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the Kafka connections page, click the options menu (three vertical dots) for the connection that you want to delete and click **Delete**.
5. Enter the name of the connection to confirm that you want to delete your connection. Then, click **Delete**.

Your connection is deleted.

Creating LogDNA connections

A LogDNA connection provides log data, which is used to establish a baseline of normal behavior and then identify anomalies. These anomalies can be correlated with other alerts and events, and published to your ChatOps interface to help you determine the cause and resolution of a problem.

For more information about working with LogDNA connections, see the following sections:

- Creating LogDNA connections
- Enabling LogDNA connections
- Editing LogDNA connections
- Deleting LogDNA connections

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating LogDNA connections

The LogDNA connection type collects data from a LogDNA data source.

About this task

Before you create a connection, gather the following information:

- **Load**: To prevent the connection from placing an inordinate load on your data source and potentially impacting your logging operations, the connection connects to only one API with a default data frequency of 60 seconds. This restriction is controlled by using the **Sampling rate** setting in the Procedure section.
- **Access:** LogDNA data sources are cloud-based and use a service key approach to enable access.
- **Data volume**: Typical data volumes are as follows:
    - 5.5 MB (15000 lines) for 20 seconds on Development environment.
    - 49 logs (19 KB) in a 10-second interval. Pulled within 30 seconds.
    - 10 logs (3.2 KB) in a 10-second interval. Pulled within 30 seconds.
    - 195 logs (75 KB) in a 30-second interval. Pulled within 60 seconds.

Procedure

To create a LogDNA connection, step through the following sections:

- Adding a connection
- Specifying connection parameters
- Specifying field mapping
- Specifying how log data is collected for AI training

Adding a connection

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.
4. From the list of available connections, find and click the **LogDNA** tile.

   Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Connect**.

Specifying connection parameters

1. On the *Add connection* page enter the following connection information:
    - **Name**: The display name of your connection.

- **Description**: An optional description for the connection.
- **URL**: The hostname or IP address of the LogDNA API. To find the URL, log in to your LogDNA account or check the LogDNA instance in your IBM Cloud account. For more information, see the Log Analysis documentation about Endpoints.
- **Service key**: The LogDNA service key. For more information, see Service keys.
- **Certificate**: An optional certificate used to verify the SSL/TLS connection to the REST service.
- **Filters**: Defines subsets of data that is pulled from LogDNA. For example, **`healthcheck (successful OR ping)`** returns all of the log lines with the word **`healthcheck`** and without the word **`successful`**, and all of the log lines with the word **`healthcheck`** and with the word **`ping`**. For more information about LogDNA filtering, see LogDNA filtering documentation.
- **Base parallelism**: Select a value to specify the number of Flink jobs that can run in parallel. These jobs run to process and normalize the collected data. The default value is 1. However, it is recommended to use a higher value than 1 so that you can process data in parallel. This value cannot exceed the total available free Flink slots. In a small environment, the available flinks slots are 16, while in a large environment, the maximum available slots are 32. If you are collecting historical data with this connection, you can set this value to be equal to the **source** parallelism.
- **Sampling rate**: The rate at which data is pulled from the live source (in seconds). The default value is **`60`**.
- **JSON processing option**: Select a JSON processing option.
  - **None**: The default option. The JSON is not processed or modified.
  - **Flatten**: This option flattens the JSON object by removing the opening and closing braces.
  - **Filter**: This option extracts the JSON object and replaces it with an empty string.
  - For more information about the options, see Managing embedded JSON.

2. Click **Test connection**.

   Note: To improve data throughput, you can increase the base parallelism value incrementally. For more information about maximum base parallelism for starter and production deployments, see Improving data streaming performance for log anomaly detection.

3. Click **Next** to move to the next page.

Specifying field mapping

1. On the *Field mapping* page you can improve search performance by mapping the fields from your implementation fields to standard fields within IBM Cloud Pak for AIOps.
   - For more information about how field mappings are defined, see Mapping data from incoming sources.
   - For more information about using mappings to clean your data for use in IBM Cloud Pak for AIOps, see Cleaning mapped data using regular expressions.

   The following code snippet displays an example of field mapping by using the supported format. When coding your mapping, use this example to help you.

```
{
  "codec": "logdna",
  "rolling_time": 10,
  "instance_id_field": "_app",
  "log_entity_types": "container",
  "message_field": "_line",
  "timestamp_field": "_ts"
}
```

   You can also map from a nested field in the source data to a field in IBM Cloud Pak for AIOps data. For example, in the following code snippet, the line **`"message_field": "_line.message",`** maps the field **`message_field`** in IBM Cloud Pak for AIOps data to the nested field **`_line.message`** in the LogDNA data.

```
{
 "rolling_time": 10,
 "instance_id_field": "_app",
 "log_entity_types": "container",
 "message_field": "_line.message",
 "timestamp_field": "_ts"
}
```

You can also create an OR relationship between multiple nested fields, where the OR relationship is expressed by using a semicolon. For example, in the following code snippet, the line **`"message_field": "_line.message1;_line.message2;_line.message3",`** maps the field **`message_field`** in IBM Cloud Pak for AIOps data to one of three possible nested fields in the LogDNA data:

- **`_line.message1`**, or
- **`_line.message2`**, or
- **`_line.message3`**
    - The system first looks for **`_line.message1`** in the LogDNA data record, and if it finds this field, then it uses it for the mapping.
    - If it can't find **`_line.message1`** in the LogDNA data record, then it looks for the next item **`_line.message2`** and if it finds it, then it uses it for the mapping.
    - If it can't find **`_line.message2`** in the LogDNA data record, then it looks for the next item **`_line.message3`** and if it finds it, then it uses it for the mapping.

```
{
  "rolling_time": 10,
  "instance_id_field": "_app",
  "log_entity_types": "container",
  "message_field": "_line.message1;_line.message2;_line.message3",
  "timestamp_field": "_ts"
}
```

2. Click **Next** to move to the next page.

Specifying how log data is collected for AI training

1. On the *AI training and log data* page, select how you want to manage collecting data for use in AI training and anomaly detection. Click the *Data flow* toggle to turn on data flow and then select how you want to collect data:
    - **Live data for continuous AI training and anomaly detection**: A continuous collection of data from your connection is used to both train AI models and analyze your data for anomalous behavior.

      Note: After an initial installation, there is no data at all in the system. If you select this option, then the two different log anomaly detection algorithms behave in the following ways:

        - Natural language log anomaly detection does not initially detect anomalies as no model has been trained. You can retrieve historical data (select **Historical data for initial AI training**) to speed up the retrieval of data to train on, or you can leave the **Live data for continuous AI training and anomaly detection** setting on. In the latter case, the system gathers training data live and after a few days there is enough data to train a model. When this model is deployed, then it detects anomalies as normal.
        - Statistical baseline log anomaly detection does not detect anomalies for the first 30 minutes of data collection. This is because it does not have a baseline yet. After 30 minutes of live data collection the baseline is automatically created. After that it detects anomalies on an ongoing basis, while continuing to gather data and improve its model every 30 minutes.

    - **Live data for initial AI training**: A single set of training data used to define your AI model. Data collection takes place over a specified time period that starts when you create your connection.

      Note: Selecting this option causes the system to continue to collect data while the option is enabled; however, the data is collected for training only, and not for log anomaly detection. For more information about AI model training, including minimum and ideal data quantities, see Configuring AI training.

- **Historical data for initial AI training**: A single set of training data used to define your AI model. You must specify a Start and End date, and the parallelism of your source data. Historical data is harvested from existing logs in your connection over a specified time period in the past.
    - **Start date**: Select a start date from the calendar and enter the time in hh:mm (hours and minutes) format.

      Note: The start date must not exceed 31 days from the present as the maximum time period for historical data collection is 31 days. The recommended time period is two weeks.
    - **Time zone**: Select your time zone from the dropdown list.
    - **End date and time**: Click **Add end date** and select an end date from the calendar and enter the time in hh:mm format.

      Note: If you do not specify the end date, then live data collection follows the historical data collection. If you do not want to set an end date, click **Remove end date**.
    - **Source parallelism (1-50)**: Select a value to specify the number of requests that can run in parallel to collect data from the source. Generally, you can set the value to equal the number of days of datat that you want to collect. When you are setting this value, consider the number of requests that are allowed by the source in a minute. For example, if only 1-2 requests are allowed, set the value to be low.

Important: Keep in mind the following considerations when you select your data collection type:

- Anomaly detection for your connection occurs if you select *Live data for continuous AI training and anomaly detection*.
- Different types of AI models have different requirements to properly train a model. Make sure that your settings satisfy minimum data requirements. For more information about how much data you need to train different AI models, see Configuring AI training.

2. Click **Done**.

You have created a LogDNA connection in your instance. After you create your connection, you must enable the data collection to connect your connection with the AI of IBM Cloud Pak for AIOps. For more information about enabling your connection, see Enabling LogDNA connections.

To create more connections (such as a ChatOps connection), see Configuring data and tool connections.

For more information about working with the insights provided by your connections, see ChatOps insight management.

Enabling and disabling LogDNA connections

If you didn't enable your data collection during creation, you can enable your connection afterward. You can also disable a previously enabled connection the same way. If you selected *Live data for initial AI training* when you created your connection, you must disable the connection before AI model training. To enable or disable a created connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Manage connections* tab of the *Data and tool connections* page, click the **LogDNA** connection type.
4. Click the connection that you want to enable or disable.
5. Go to the *AI training and log data* section. Set **Data connection** to **On** or **Off** to enable or disable data collection. Disabling data collection for a connection does not delete the connection.

You enabled or disabled your connection. For more information about deleting a connection, see Deleting LogDNA connections.

Editing LogDNA connections

After you create your connection, your can edit the connection. For example, if you specified *Historical data for initial AI training* but now want your connection to pull in live data for continuous monitoring, you can edit it. To edit a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **LogDNA** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the LogDNA connections page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.
5. Edit your connection as required. Click **Save** when you are done editing.

Your connection is now edited. If your application was not previously enabled or disabled, you can enable or disable the connection directly from the interface. For more information about enabling and disabling your connection, see Enabling and disabling LogDNA connections. For more information about deleting a connection, see Deleting LogDNA connections.

Deleting LogDNA connections

If you no longer need your LogDNA connection and want to not only disable it, but delete it entirely, you can delete the connection from the console.

Note: You must disable data collection before you delete your connection. For more information about disabling data collection, see Enabling and disabling LogDNA connections.

To delete a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **LogDNA** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the LogDNA connections page, click the options menu (three vertical dots) for the connection that you want to delete and click **Delete**.
5. Enter the name of the connection to confirm that you want to delete your connection. Then, click **Delete**.

Your connection is deleted.

Note: If you manually send LogDNA logs from an archive, the model performs as expected only on other logs that are sent from the archive. Mixing archived data and data from the active service instance results in lower quality models.

Connecting with Metric Manager

To integrate Metric Manager (formerly IBM® Operations Analytics Predictive Insights) with IBM Cloud Pak for AIOps , first ensure that you implement the prerequisites, and then change the rules file in Metric Manager.

Note: IBM Cloud Pak for AIOps comes with built-in metric capabilities. Connect with Metric Manager only if these capabilities are insufficient to your needs.

Prerequisites

To allow the forwarding of event data from Metric Manager to IBM Cloud Pak for AIOps , and the integration of both, do the following:

1. Event forwarding from Metric Manager to IBM Cloud Pak for AIOps is done by forwarding from the on-premises objectserver to IBM Cloud Pak for AIOps by using a uni-directional gateway. To set up this gateway, follow the instructions described here.

2. For Metric Manager 1.3.6 integration with IBM Cloud Pak for AIOps , apply InterimFix005 available here.

Procedure

Change the rules file in Metric Manager. Make the following changes to the rules file (located here: $PI_HOME/probe/omnibus/probes/linux2x86/stdin-tasp.rules) in Metric Manager.

Note: Ensure you backup the rules file before you edit the file.

1. In the file, locate the following two lines:

```
#@Identifier = @Node + " " + @AlertKey + " " + @AlertGroup + " " + @Type + " " + @Agent +
" " + @Manager
@Identifier = $Identifier
```

   Remove the '#' from the first line and add it to the start of the second, so that it changes to the following:

```
@Identifier = @Node + " " + @AlertKey + " " + @AlertGroup + " " + @Type + " " + @Agent +
" " + @Manager
#@Identifier = $Identifier
```

2. Add the following before the last closing bracket "`}`", replacing "`PI_UI_HOST.example.com=`" with the fully qualified domain name of the Metric Manager UI host:

```
@URL =
"https://PI_UI_HOST.example.com:9998/predictiveinsights/jsp/wlp/wlpAnomalyView.jsp?
anomalyKey=" + $TASPIdentifier
update(@URL)
```

3. Add either of the following just after the last closing bracket "`}`":

   - IBM Cloud Pak for AIOps

```
#For integration with IBM Cloud Pak for AIOps :

if (regmatch($Class, "89350") || regmatch($Class, "89360")) {
   @AlertGroup = "ANOMALY:" + @TASPAnomalousMetricGroups + ":" +
@TASPAnomalousMetrics
   update(@AlertGroup)
   }

   ## Note: Do not forward these consolidated events from Metric Manager, as the
downstream system will do this grouping:

   if (regmatch($TASPIdentifier, "Node") || regmatch($TASPIdentifier,
"RelatedEvents") || regmatch($TASPIdentifier, "Incident")) {
  discard
  }
```

   - IBM Cloud Pak for AIOps Event Manager

```
#For integration with IBM Cloud Pak for AIOps Event Manager

if (regmatch($Class, "89350") || regmatch($Class, "89360")) {
   @AlertGroup = "ANOMALY:" + @TASPAnomalousMetricGroups + ":" +
@TASPAnomalousMetrics
   update(@AlertGroup)
   }
   $Identifier = regreplace($Identifier, "(_\d{13}$)", "" )
   @Identifier = $Identifier
   update(@Identifier)

   ## Note: Do not forward these consolidated events from Metric Manager, as the
downstream system will do this grouping:

   if (regmatch($TASPIdentifier, "Node") || regmatch($TASPIdentifier,
"RelatedEvents") || regmatch($TASPIdentifier, "Incident")) {
  discard
  }
```

4. To start the gateway, run the following:

```
cd /opt/IBM/tivoli/netcool/omnibus/bin
nohup ./nco_g_objserv_uni -name NCO_GATE -propsfile
/opt/IBM/tivoli/netcool/omnibus/gates/aiops/aiops.props &
```

Creating IBM Tivoli Netcool/OMNIbus connections

You can send event data from an existing IBM Tivoli Netcool/OMNIbus environment to IBM Cloud Pak for AIOps through a Netcool connection. This connection connects the IBM Tivoli Netcool/OMNIbus ObjectServer with IBM Cloud Pak for AIOps.

With this integration, you can complete the following tasks:

- You can see IBM Tivoli Netcool/OMNIbus event data in the Alert Viewer in IBM Cloud Pak for AIOps. Events are converted into alerts on the IBM Cloud Pak for AIOps side.
- You can take advantage of the IBM Cloud Pak for AIOps AI features. For example, your event data is correlated into incidents, and you can associate topology with your data. You can also create runbooks to automate operations based on your data.

For more information about working with Netcool connections, see the following sections:

- About IBM Tivoli Netcool/OMNIbus connections
- Before you begin
- Creating Netcool connections
- Editing Netcool connections
- Deleting Netcool connections

About IBM Tivoli Netcool/OMNIbus connections

- Though a Netcool connection is bidirectional, not all data is synchronized. For example, if you close an alert in IBM Cloud Pak for AIOps, the alert does not get closed on the IBM Tivoli Netcool/OMNIbus ObjectServer. The **`AIOpsState`** column in the ObjectServer gets updated to closed but the alert continues to stay open on the ObjectServer. The operator must write a trigger on the ObjectServer to close such alerts. However, if an alert is closed on the ObjectServer, the alert does get closed in IBM Cloud Pak for AIOps.

- If you plan to create multiple IBM Tivoli Netcool/OMNIbus connections, the set of events from each connection must be unique and not overlap with other connections. For example, if you want to retrieve two separate sets of events, such as odd and even ServerSerial events, then you must use two different connections. The connection filter helps to group events that you want to synchronize. For more information about the connection filter, see Filter clause.

    - If you have multiple ObjectServer pairs, you must create a corresponding Netcool connection for each pair. Each ObjectServer pair requires a separate Netcool connection. Do not reconfigure an existing Netcool connection to access other ObjectServer pair.

      Note: An ObjectServer pair must synchronize with only one IBM Cloud Pak for AIOps instance through one or more Netcool connections. If you want to repoint an ObjectServer pair to a different IBM Cloud Pak for AIOps instance, clear the **`AIOpsAlertID`** and **`AIOpsState`** columns in the **`alerts.status`** table on the ObjectServer before you start the new deployment.

    - Parallel Netcool connections can be useful for distributing event load. However, use such a deployment only if necessary.

Before you begin

To configure a Netcool connection, you need the following information:

- ObjectServer credentials
- Hostname and port for the primary and backup ObjectServers
- Custom ObjectServer column mapping

Creating Netcool connections

To create a Netcool connection for IBM Tivoli Netcool/OMNIbus, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.
4. From the list of available connections, find and click the **Netcool** tile.

   Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Connect**.
6. To configure the ObjectServers with IBM Cloud Pak for AIOps columns, complete the following steps under *Configure ObjectServer*:

   - Open a command line.
   - Connect to the Netcool ObjectServer Host.
   - Save the following SQL commands in a file called *add_trigger.sql*:

     ```
     -- Add IBM Cloud Pak for AIOps required fields ---
     alter table alerts.status add column AIOpsAlertId varchar(64);
     go

     alter table alerts.status add column AIOpsState varchar(16);
     go

     -- Add the Netcool Connector automation trigger group --
     create or replace trigger group NetcoolConnectorAutomation;
     go

     -- Enable the trigger --
     alter trigger group NetcoolConnectorAutomation set enabled true;
     go

     -- Add the clear automation trigger --
     create or replace trigger netcool_connector_clear_alerts
     group NetcoolConnectorAutomation
     priority 1
     comment 'Netcool Connector clear alerts that are closed in IBM Cloud Pak for AIOps.'
     every 10 seconds
     when get_prop_value('ActingPrimary')%='TRUE'
     begin
        -- Set Sev to 0  if AIOpsState is closed
        for each row problem in alerts.status where
            problem.Type = 1 and problem.Severity > 0 and problem.AIOpsAlertId != '' and
     problem.AIOpsState = 'closed'
        begin
            update alerts.status via problem.Identifier set Severity = 0;
        end;
     end;
     go
     ```

     Note: The ScopeID column is available by default on Red Hat OpenShift Container Platform. However, use the following command for setting it up for on-premises ObjectServer:

     ```
     alter table alerts.status add column ScopeID varchar(255);
     go
     ```

   - Run the following command to install the required columns and trigger by using the *add_trigger.sql* file:

     ```
     $OMNIHOME/bin/nco_sql -server <object_server_name> -username root -password
     <password> < add_trigger.sql
     ```

7. Click **Next**.
8. Enter the following *Add connections* information:
    - **Name**: The display name of your connection.
    - **Description**: An optional description for the connection.
    - **Deployment options**: Select **Local** or **Remote** option.

      Note: If you select local, the deployment is in the same cluster where IBM Cloud Pak for AIOps is installed. If you select remote, the deployment happens wherever you choose, for example, on a different network region, on Saas, or remotely on premises (VM, cluster, or container).
9. Fill the *ObjectServer pair* fields. For more information about the fields and values to be entered, see Table 1.
10. Click **Test Connection**. If successful, a *Test succeeded* message is displayed. If it fails, one of the following reasons appears:

- Filter is not valid
- Unable to connect to ObjectServer
- IBM Cloud Pak for AIOps fields not present

Note: If the test connection fails, check:

- if the *TLS certificate* is correct.
- if the *NetworkPolicy* is created with the correct ports and in the correct namespace.
- if the *ExternalName* service is created with correct service name, ports, and in the correct namespace.

For more information about *TLS certificate*, *NetworkPolicy*, and *ExternalName* service, see TLS-enabled connection.

11. Click **Next**.
12. Enter the following *Collect event data* information to define how active IBM Tivoli Netcool/OMNIbus events are collected.
    - **Enable data collection**: This toggle is set to *on* by default. It starts to collect event data from IBM Tivoli Netcool/OMNIbus immediately on creation of the connection.

- **Mapping**: This field provides mapping for standard IBM Tivoli Netcool/OMNIbus event columns that use Jsonata syntax. You can add your own custom ObjectServer columns to this mapping. If they have custom alert columns in your ObjectServer, you must add them to the Jsonata mapping provided in the Mapping textbox.

  Note: To customizing the mapping, copy the text out of the mapping textbox and into your preferred text editor. Then, add the mapping for those custom columns there and check the code syntax. When done, copy the Jsonata code back into the mapping textbox. If successful, a *Valid Jsonata configuration* message is displayed.

13. Click **Done**.

You have created a Netcool connection in your instance, and a message appears on the connections page that states *Connection completed. IBM Cloud Pak for AIOps has successfully processed your request*.

After you integrate with IBM Tivoli Netcool/OMNIbus, use IBM Cloud Pak for AIOps and the IBM Cloud Pak for AIOps UI for AI and visualization features. Use IBM Tivoli Netcool/OMNIbus and the associated UI tools for obtaining and sending data to IBM Cloud Pak for AIOps.

Notes:

- If you are integrated and are not yet ready, or able, to move to using IBM Cloud Pak for AIOps and the IBM Cloud Pak for AIOps UI fully. You can continue to use the IBM Tivoli Netcool/OMNIbus DASH custom dashboards and Web GUI event viewer as you transition to IBM Cloud Pak for AIOps. If you need to continue to use IBM Tivoli Netcool/OMNIbus tools, you can configure launch-in-context links from the Web GUI to the IBM Cloud Pak for AIOps UI to help users access and use IBM Cloud Pak for AIOps UI. For more information, see Configuring launch-in-context from Netcool
- If you plan to connect with an IBM Tivoli Netcool/OMNIbus deployment in an environment that is enabled to support Federal Information Processing Standards (FIPS), you need to configure your connection to also enable the support. For more information, see Enabling support for Federal Information Processing Standards (FIPS).

ObjectServer pairs

The following table outlines the ObjectServer pair fields and their values.

Table 1. ObjectServer pair fields

ObjectServer pair fields	Description	Values
Admin username	Enter a username to authenticate to the ObjectServer.	$ NOI_OMNI_USERNAME
Admin password	Enter a password to authenticate to the ObjectServer.	$ NOI_OMNI_PASSWORD
Filter (optional)	Define the alerts that you want to synchronize by using ObjectServer SQL where clause syntax. Without filter, the connection takes all alerts from the database table.	For example, Severity > 4 or Manager LIKE 'CEM'. For more examples, see Filter clause.
Is TLS enabled?	Enable or disable the toggle button. For more information about TLS-enabled connection, see TLS-enabled connection.	**Yes** or **No**
Primary ObjectServer hostname	Enter a hostname for the primary ObjectServer connection.	$ NOI_PROXY_PRIMARY_SVC
Primary ObjectServer port	Specify a port.	$ NOI_PROXY_PRIMARY_PORT
Backup ObjectServer hostname (Optional)	Enter a hostname for the backup ObjectServer connection.	$ NOI_PROXY_BACKUP_SVC
Backup ObjectServer port (Optional)	Specify a port.	$ NOI_PROXY_BACKUP_PORT
Trusted certificates (if TLS is enabled)	Enter the TLS certificate that is used to verity the SSL/TLS connection to the ObjectServers.	See TLS-enabled connection

TLS-enabled connections

A TLS-enabled connection provides end-to-end security to data that is transmitted through it. To activate a TLS-enabled connection, you must provide a Transport Layer Security (TLS) certificate. A TLS certificate is a type of digital certificate, issued by a certificate authority (CA), verifying the ownership of the domain name.

You can create a TLS-enabled connection to either Red Hat® OpenShift® Container Platform ObjectServer, or the On-premises IBM Tivoli Netcool/OMNIbus ObjectServer. For more information, see:

- Creating a TLS-enabled connection to a Red Hat® OpenShift® Container Platform ObjectServer

Editing Netcool connections

After you create your connection, you can edit it. To edit a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Netcool** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the Netcool connections page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.
5. Edit your connection.

   Note: While you can modify all other fields, you cannot alter the actual connection name at this stage, as it is already created and thus shows as read-only.

6. Click **Save** when you are done editing.

Your connection is edited. If you did not previously enable or disable your application, you can enable or disable the connection directly from the console. For more information about deleting a connection, see Deleting Netcool connections.

Deleting Netcool connections

If you no longer need your Netcool connection and want to not only disable it, but delete it entirely, you can delete the connection from the console.

Note: You must disable data collection before deleting your connection.

To delete a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Netcool** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the Netcool connections page, click the options menu (three vertical dots) for the connection that you want to delete and click **Delete**.
5. Enter the name of the connection to confirm that you want to delete your connection. Then, click **Delete**.

Your connection is deleted.

If you want to remove columns and automation triggers from Netcool ObjectServer, complete the following steps:

1. Open a terminal.
2. Connect to the Netcool ObjectServer host.
3. Save the following SQL commands into a file called *remove_trigger.sql*:

```
alter trigger group NetcoolConnectorAutomation set enabled false;
go

drop trigger netcool_connector_clear_alerts;
go

drop trigger group NetcoolConnectorAutomation;
go

alter table alerts.status drop column AIOpsAlertId;
go

alter table alerts.status drop column AIOpsState;
go
```

4. Run the following command to install the required columns and trigger by using the remove_trigger.sql file:

```
$OMNIHOME/bin/nco_sql -server <object_server_name> -username root -password <password> <
remove_trigger.sql
```

Configuring launch-in-context from IBM Tivoli Netcool/OMNIbus

If you have integrated IBM Cloud Pak for AIOps and IBM Tivoli Netcool/OMNIbus, you can configure launch-in-context links from the Web GUI Event Viewer and the main DASH navigation to the IBM Cloud Pak for AIOps UI to help your users access and use the IBM Cloud Pak for AIOps UI.

With these links, your users can use the Web GUI Event Viewer and DASH to manage the gathering of event data, which is then sent to IBM Cloud Pak for AIOps. These users can then launch the IBM Cloud Pak for AIOps UI for AI analysis and visualization of the data.

Creating a connection between IBM Cloud Pak for AIOps and IBM Tivoli Netcool/OMNIbus to send event data to Cloud Pak for AIOps is prerequisite. For more information, see Creating IBM Tivoli Netcool/OMNIbus connections.

For more information about configuring launch-in-context, see:

- Adding incidents and alerts to the DASH navigation menu
- Creating tools for launch-in-context from Netcool/OMNIbus Event Viewer

Adding incidents and alerts to the DASH navigation menu

If you have integrated IBM Cloud Pak for AIOps and IBM Tivoli Netcool/OMNIbus, you can add a launch-in-context link to the IBM Cloud Pak for AIOps Incidents and alerts page to the DASH navigation menu for IBM Tivoli Netcool/OMNIbus.

Procedure

1. Navigate to the JAZZSM_HOME directory and open the following file:
   <JAZZSM_HOME>/profile/config/cells/JazzSMNode01Cell/applications/isc.ear/deployments/isc/isclite.war/WEB-INF/navigation.xml

2. At the end of the content wrapped between the following tags:

   ```
   <navigation:navigation-tree> {...} </navigation:navigation-tree>
   ```

   Append the following XML snippet, replacing the values marked as **<CP4WAIOPS_STORIES_URL>** and **<CP4WAIOPS_ALERTS_URL>** with valid URLs to the Cloud Pak for AIOps Incidents and Alerts pages:

   ```
   <navigation:nav-element formPersistence="false" hidden="false" iconClass="cloudIcon"
   isWscNode="false" layoutUpdatedAtRuntime="false"
   moduleID="com.ibm.isclite.global.custom.module"
   moduleIDOrig="com.ibm.isclite.global.custom.module" nodeType="label"
   securityUpdatedAtRuntime="false" stateMaintained="false" taskType="hiddenTask"
   uniqueName="com.ibm.isclite.global.custom.module-SPSVS-label.W8DQOK4vA5VBn4RGGldV-
   u11650881799360" wscRole="all authenticated portal users" wscRoleType="User">
     <navigation:title>
       <base:nls-string lang="en">CP4WAIOps</base:nls-string>
     </navigation:title>
     <navigation:parentTree parentTreeRef="root"/>
     <navigation:parameter name="com.ibm.console.resource.previewImage">
       <base:value>
       <base:string>CP4WAIOps</base:string>
       </base:value>
     </navigation:parameter>
     <navigation:parameter name="com.ibm.console.resource.description">
       <base:value>
       <base:string>CP4WAIOps</base:string>
       </base:value>
     </navigation:parameter>
     <navigation:page-params/>
     <navigation:nav-element formPersistence="false" hidden="false" isWscNode="false"
   layout-element-ref="com.ibm.isclite.admin.Freeform.layoutElement.A__1650882838344"
   layoutUpdatedAtRuntime="true" moduleID="com.ibm.isclite.global.custom.module"
   moduleIDOrig="com.ibm.isclite.global.custom.module" nodeType="url"
   securityUpdatedAtRuntime="false" stateMaintained="true" taskType="multiTask"
   uniqueName="com.ibm.isclite.global.custom.module-SPSVS-
   com.ibm.isclite.admin.Freeform.navigationElement.pagelayoutA.modified.9L7ci90Q2Z7x3uGKVJ2
   v5uh1650882838339" url="<CP4WAIOPS_STORIES_URL>" wscRole="iscadmins"
   wscRoleType="Editor">
       <navigation:title>
       <base:nls-string lang="US">Incidents</base:nls-string>
       </navigation:title>
       <navigation:parentTree parentTreeRef="label.W8DQOK4vA5VBn4RGGldV-u11650881799360"/>
     </navigation:nav-element>
     <navigation:nav-element formPersistence="false" hidden="false" isWscNode="false"
   layout-element-ref="com.ibm.isclite.admin.Freeform.layoutElement.A__1651490860014"
   layoutUpdatedAtRuntime="true" moduleID="com.ibm.isclite.global.custom.module"
   moduleIDOrig="com.ibm.isclite.global.custom.module" nodeType="url"
   ```

```
securityUpdatedAtRuntime="false" stateMaintained="true" taskType="multiTask"
uniqueName="com.ibm.isclite.global.custom.module-SPSVS-
com.ibm.isclite.admin.Freeform.navigationElement.pagelayoutA.modified.taRbaKSlGPLajMfN2Bb
RVfB1651490860008" url="<CP4WAIOPS_ALERTS_URL>" wscRole="iscadmins" wscRoleType="Editor">
    <navigation:title>
    <base:nls-string lang="US">Alerts</base:nls-string>
    </navigation:title>
    <navigation:parentTree parentTreeRef="label.W8DQOK4vA5VBn4RGGldV-u11650881799360"/>
  </navigation:nav-element>
</navigation:nav-element>
```

3. Save the `navigation.xml` file.
4. Log out of DASH, then log back in again.

Creating a tool for launch-in-context from IBM Tivoli Netcool/OMNIbus Event Viewer

To use a launch-in-context tool, you must add it to the list of available IBM Tivoli Netcool/OMNIbus tools by using the **Tool Configuration** option within the IBM Tivoli Netcool/OMNIbus Web GUI. You must then add the tool to the list of tools that appear on the pop-up menu in the IBM Tivoli Netcool/OMNIbus Event Viewer by using the **Menu Configuration** option.

Complete the steps in the following procedure to create the tool for launch-in-context from selected events in the Event Viewer.

Procedure

1. As a user with the admin role, log in to your DASH web application.
2. Open the Netcool/OMNIbus Web UI tool configuration page: **Administration > Event Management Tools > Tool Configuration**.
3. Click the create tool icon to create a new tool.
4. Enter a name for the tool, for example "View in IBM Cloud Pak for AIOps".
5. From the **Type** drop-down field, select **CGI/URL**.
6. Paste the following URL in the field provided:

   ```
   <aiops hostname>:<aiops port>/aiops/cfd95b7e-3bc7-4006-a4a8-a73a79c71255/resolution-
   hub/alerts?selectedRowIds=alerts.status.undefined:
   {$selected_rows.AIOpsAlertId}&selectedColumnId=summary
   ```

   Where `<aiops hostname>:<aiops port>` is the URL of your IBM Cloud Pak for AIOps instance in the format of protocol://hostname:port.
7. Click **Save**.
8. Open the Netcool/OMNIbus Web UI menu configuration page: **Administration > Event Management Tools > Menu Configuration**.
9. Click the **Alerts** menu and select **Modify**.
10. Locate the tool that you created in the available list and add it to the list of selected tools.
11. Click **Save**.

This tool should now be available in the Event Viewer when you right-click on an Event. Selecting this tool launches into the Cloud Pak for AIOps alerts for the selected event.

Creating a TLS-enabled IBM Tivoli Netcool/OMNIbus connection to an Red Hat OpenShift ObjectServer

If you are connecting IBM Cloud Pak for AIOps to IBM Tivoli Netcool/OMNIbus and you have TLS enabled for your Red Hat OpenShift ObjectServer, you need to extract the TLS certificate to add to IBM Cloud Pak for AIOps.

A TLS-enabled connection between IBM Cloud Pak for AIOps and IBM Tivoli Netcool/OMNIbus provides end-to-end security to data that is transmitted through it. To create a TLS-enabled connection to a Red Hat OpenShift ObjectServer, complete the following steps:

1. Get the host and port for primary and backup ObjectServers from the TLS Proxy service:

   ```
   NOI_INSTANCE=<NOI_instance_name>
   NOI_NAMESPACE=<NOI_namespace>
   NOI_PROXY_SVC=${NOI_INSTANCE}-proxy
   NOI_PROXY_PRIMARY_SVC=${NOI_PROXY_SVC}.${NOI_NAMESPACE}.svc
   NOI_PROXY_PRIMARY_PORT=$(oc describe svc -n $NOI_NAMESPACE $NOI_PROXY_SVC | grep aggp-
   proxy-port | awk -F' ' '{ print $3 }' | awk -F'/' '{ print $1 }')
   NOI_PROXY_BACKUP_SVC=${NOI_PROXY_SVC}.${NOI_NAMESPACE}.svc
   NOI_PROXY_BACKUP_PORT=$(oc describe svc -n $NOI_NAMESPACE $NOI_PROXY_SVC | grep aggb-
   proxy-port | awk -F' ' '{ print $3 }' | awk -F'/' '{ print $1 }')
   ```

   Where, **<NOI_instance_name>** is the Netcool instance name that is installed in the cluster and **<NOI_namespace>** is the namespace where Netcool is installed. For example, **NOI_NAMESPACE=noi**.

2. Create a *NetworkPolicy* in the Netcool namespace to allow the probe to access the TLS Proxy service and port in NOI namespace. To create the *NetworkPolicy*, complete the following steps:

   1. Set the environment variables.

      ```
      NOI_INSTANCE=<NOI_instance_name>
      NOI_NAMESPACE=<NOI_namespace>
      NOI_OBJSERV_AGG_PRIMARY_SVC=$NOI_INSTANCE-objserv-agg-primary
      NOI_OBJSERV_AGG_BACKUP_SVC=$NOI_INSTANCE-objserv-agg-backup
      NOI_OBJSERV_AGG_PRIMARY_PORT=$(kubectl get svc -n $NOI_NAMESPACE
      $NOI_OBJSERV_AGG_PRIMARY_SVC -o jsonpath='{.spec.ports[?(@.name=="primary-tds-
      port")].port}')
      NOI_OBJSERV_AGG_IDUC_PORT=$(kubectl get svc -n $NOI_NAMESPACE
      $NOI_OBJSERV_AGG_PRIMARY_SVC -o jsonpath='{.spec.ports[?(@.name=="primary-iduc-
      port")].port}')
      ```

      Where **<NOI_instance_name>** is the Netcool instance name that is installed in the cluster and **<NOI_namespace>** is the namespace where Netcool is installed.

   2. Run the following commands:

      ```
      cat << EOF | tee >(oc apply -f -) | cat
      kind: NetworkPolicy
      apiVersion: networking.k8s.io/v1
      metadata:
         name: noi-allow-objectserver
         namespace: ${NOI_NAMESPACE}
      spec:
         podSelector:
            matchLabels:
               app.kubernetes.io/instance: ${NOI_INSTANCE}
               appClass: objserv
         ingress:
            - ports:
               - protocol: TCP
                 port: ${NOI_OBJSERV_AGG_PRIMARY_PORT}
               - protocol: TCP
                 port: ${NOI_OBJSERV_AGG_IDUC_PORT}
         policyTypes:
            - Ingress
      EOF
      ```

Note: Check to see whether the existing *NetworkPolicy* allows the probe to access the TLS Proxy service and port in Netcool namespace. The *NetworkPolicy* must show both the primary and IDUC ports.

3. Create an *ExternalName* service for the primary and IDUC ports. It maps the namespace scoped service to the Netcool service in the namespace. To create the *ExternalName* service, complete the following steps:

    1. Set the environment variables.

    ```
    AIMANAGER_NAMESPACE=<AIManager_namespace>
    oc project $AIMANAGER_NAMESPACE
    ```

    Where **<AIManager_namespace>** is the namespace where IBM Cloud Pak for AIOps is installed.

    2. Run the following commands.

    ```
    cat << EOF | tee >(oc apply -f -) | cat
    apiVersion: networking.k8s.io/v1
    kind: Service
    apiVersion: v1
    metadata:
      name: ${NOI_INSTANCE}-objserv-agg-primary
      namespace: ${AIMANAGER_NAMESPACE}
    spec:
      type: ExternalName
      externalName: ${NOI_INSTANCE}-objserv-agg-
    primary.${NOI_NAMESPACE}.svc.cluster.local
      ports:
      - port: 4101
        name: primary-iduc-port
    ---
    kind: Service
    apiVersion: v1
    metadata:
      name: ${NOI_INSTANCE}-objserv-agg-backup
      namespace: ${AIMANAGER_NAMESPACE}
    spec:
      type: ExternalName
      externalName: ${NOI_INSTANCE}-objserv-agg-
    backup.${NOI_NAMESPACE}.svc.cluster.local
      ports:
      - port: 4101
        name: backup-iduc-port
    ---
    EOF
    ```

4. Obtain a TLS certificate. To obtain the TLS certificate for the ObjectServer, run the following command:

```
oc extract secret/$NOI_INSTANCE-proxy-tls-secret -n $NOI_NAMESPACE   --to=. --
keys=tls.crt
```

Filter clauses for an IBM Tivoli Netcool/OMNIbus connection

For a connectiong between IBM Cloud Pak for AIOps and IBM Tivoli Netcool/OMNIbus, you can use the filter clause to define the alerts that you want to send to IBM Cloud Pak for AIOps. You can create filter clauses that are suitable for your environment by using an ObjectServer SQL **where** clause syntax.

To generate filter clauses, use fields from the ObjectServer **alerts.status** table, such as **ServerName**, **Agent**, or **Manager** for the syntax. The following example shows this syntax:

```
ServerName IN ('Server1','Server2')
```

Filter clause examples

The following table outlines some of the examples of **FILTER** clause and their description:

Filter clause examples

Filter examples	Description
`Agent != 'OMNIbus SelfMonitoring'`	This filter clause filters all events in the ObjectServer `alerts.status` table with the `Agent` field not equal to `OMNIbus SelfMonitoring`.
`Manager not like 'OMNIbus Self Monitoring*'`	This filter clause filters all events in the ObjectServer `alerts.status` table with the `Manager` field not starting with `OMNIbus Self Monitoring`. For more examples of `Manager` field, see Note.
`Severity >= 2`	This filter clause filters all events in the ObjectServer `alerts.status` table with `Severity` field equal to or beyond the warning level. Thus, the connection selects only those alerts that have severity more than or equal to 2. For more information about severity level representation, see Event severity levels.
`ServerName IN ('AGG_P','AGG_B')`	This filter clause filters all events in the ObjectServer `alerts.status` table with ServerName equal to `AGG_P` and `AGG_B`. Thus, the connection selects only those alerts that originate from `'AGG_P'` and `'AGG_B'`.

The following are some examples of the `Manager` field:

```
OMNIbus Self Monitoring @AGG_P
OMNIbus Self Monitoring @AGG_B
```

Where `AGG_P` and `AGG_B` are server names.

You can use Boolean operators such as `AND, OR` to join conditions in the filter clause syntax. For example,

```
Acknowledged>0 AND ServerName IN ('NCOMS_US','NCOMS_CA')
```

Creating New Relic connections

A New Relic connection collects live and historical metric data from the New Relic server in IBM Cloud Pak® for AIOps. This metric data helps to monitor and analyze application performance. To activate the connection, you must have an endpoint and an access token. The metric anomaly detection AI algorithm is used to analyze the New Relic metrics for identifying deviations from the normal behavior and detecting anomalies. New Relic v6.5.2 with v2 API is the version that is supported for the New Relic connector.

For more information about working with New Relic connections, see the following sections:

- Creating New Relic connections
- Enabling and disabling New Relic connections
- Editing New Relic connections
- Deleting New Relic connections

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating New Relic connections

To create a New Relic connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.
4. From the list of available connections, find and click the **New Relic** tile.

   Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Connect**.

6. On the *Add connection* page, define the general connection details:
    - **Name**: Enter the display name of your connection.
    - **Description (optional)**: Type in a description for the connection.
    - **Endpoint**: Enter an endpoint for the connection.
    - **Access token**: Enter an access token for the connection. You can create or obtain the access token from the *API Keys* section in your New Relic account.
    - **Deployment options**: Select one of the options, **Local** or **Remote**. **Local** deploys the connection in the same cluster where IBM Cloud Pak for AIOps is installed. Alternatively, **Remote** deploys it in your chosen location.

      Note: If you select **Remote**, after adding the connection, you are provided with a bootstrap command to finish the deployment.
7. Click **Test Connection**. If successful, a Test succeeded message is displayed.
8. Click **Next**.
9. Enter the following *Collect metric data* to defines how you want to collect metric data:
    - **Enable data collection**: Set **Enable data collection** to on. However, if you disable data collection, a new connection is created, but metrics are not collected.
    - **Select application**: Select one or more applications from the list to see their associated metrics in the filter table. The filter table displays all the associated metrics of the selected applications.
    - **Search and select the required metrics from the filter table**: Search through all the metrics and select the metrics from the table. By default, all metrics are selected. To remove a selected metric, search to locate it and clear its corresponding checkbox. You can also click **Cancel** to remove all selected metrics or **Select all** to reselect them.
    - **Aggregation interval (minutes)**: Enter a value in minutes. The value represents the time interval for live or historical data collection. The default value is 5 minutes.

      Note: If aggregation interval is less than 60 minutes, IBM Cloud Pak® for AIOps collects only a maximum of past 8 days of historical data. This is because the connector aggregates data in hourly periods, instead of the time you enter, when the data request time range is older than 8 days. For more information about aggregation interval, see the New Relic documentation.
    - Select historical or live data collection. **Note**: If you choose to collect only live data, you cannot collect historical data after the live data collection starts.
        - **Live data collection**: To collect live data, complete the following steps:
            - **Only live data once the connection is established**: Select the checkbox.
            - **Scheduler run time (minutes)**: Enter a value in minutes. The value represents the frequency at which data is collected for live metrics. The default value is 5 minutes.
        - **Historical data collection**: To collect historical data, complete the following steps:
            - **Start date and time**: Select the start date from the calendar and enter the time in the *hh:mm* format (hours and minutes).

              Note: The start date must not exceed 31 days from the present as the maximum time period for historical data collection is 31 days. The recommended time period is two weeks.
            - **Time zone**: Select your time zone from the dropdown list.
            - **End date and time**: Click **Add end date** and select a date from the calendar. Also, enter the time in the *hh:mm* format (hours and minutes). If you want to remove an end date, click **Remove end date**.

Note: If you do not set an end date, IBM Cloud Pak for AIOps collects historical data from the specified start date and continues with live data collection. You cannot collect historical data after the live data collection starts.

- **Scheduler run time (minutes)**: Enter a value in minutes. The value represents the frequency at which data is collected for live metrics. The default value is 5 minutes.

  Note: Scheduler run time is only applicable to live mode. You must configure the scheduler run time only if you do not add an end date as the connector will continue to run in live mode after historical mode.

10. Click **Done**.

For more information about performance considerations for metric data collection, see Performance considerations for metric data collection.

Enabling and disabling New Relic connections

If you didn't enable your data collection during creation, you can enable your connection afterward. You can also disable a previously enabled connection the same way. To enable or disable a created connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Manage connections* tab of the *Data and tool connections* page, click the **New Relic** connection type.
4. Click the connection that you want to enable or disable.
5. Go to the *Collect metric data* section. Set **Enable data collection** to **On** or **Off** to enable or disable data collection. Disabling data collection for a connection does not delete the connection.

You enabled or disabled your connection. For more information about deleting a connection, see Deleting New Relic connections.

Editing New Relic connections

After you create your connection, you can edit it. To edit a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **New Relic** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the New Relic connections page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.
5. Edit your connection as required. Click **Save** when you are done editing.

You edited your connection.

Deleting New Relic connections

If you no longer need your New Relic connection and want to delete it entirely, you can delete the connection from the console.

To delete a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **New Relic** connection type on the *Manage connections* tab of the *Data and tool connections* page.

4. On the New Relic connections page, click the options menu (three vertical dots) for the connection that you want to delete and click **Delete**.
5. Enter the name of the connection to confirm that you want to delete your connection. Then, click **Delete**.

Your connection is deleted.

Creating observer jobs

Observers for Instana and ServiceNow are configured as part of the standard integration process on the Data and Tools Connections page in the UI. These observers can collect data from different types of data sources. Additionally, other topology observers are accessible on the Data and Tools Integrations page in the UI, and link to the Topology Manager UI. To run other observer jobs, you can configure and schedule the jobs through the Topology Manager UI.

Observers provide topology data for connections, which can help with understanding how resources are connected to each other. In addition, they provide probable cause information for events that are published to integrated collaboration tools like Slack and Microsoft Teams.

For more information about working with observer jobs, see Defining observer jobs.

Creating PagerDuty connections

A PagerDuty connection provides event data that is analyzed and processed into alerts. An incident is generated, based on an alert severity, and published to your ChatOps interface with insights into potential issues that impact operations.

Contextual alerts can be grouped into an incident by using temporal, topological, and scope correlation methods. For more information, see About events, alerts, and incidents.

The source attribute in the event is used to get the entity information. The entity information is used for the application resolution and the event grouping. For more information, see Application incidents, and Managing incidents In ChatOps.

For more information about working with PagerDuty connections, see the following sections:

- Prerequisites
- Creating PagerDuty connections
- Enabling PagerDuty connections
- Editing PagerDuty connections
- Deleting PagerDuty connections

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Prerequisites

Before you create a PagerDuty connection, you must create a ChatOps connection. For more information, see Outgoing connections.

Creating PagerDuty connections

To create a PagerDuty connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.
4. From the list of available connections, find and click the **PagerDuty** tile.

Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section.

5. On the side-panel, review the instructions and when ready to continue, click **Connect**.
6. On the *Add connection* page, define the general connection details:
    - **Name**: The display name of your connection.
    - **Description**: An optional description for the connection.
    - **API key**: The connection token for your PagerDuty instance.
    - **Service IDs**: An optional comma-separated list of PagerDuty service IDs. The service ID can be obtained from the PagerDuty URL. If you view a specific service, the PagerDuty URL might be something like the following URL: https://xyz.pagerduty.com/service-directory/P4FY6G2 In this example, the service ID is `P4FY6G2`.
    - **Team IDs**: An optional comma-separated list of PagerDuty team IDs. If you view a specific team, the PagerDuty URL might be something like the following URL: https://ibm.pagerduty.com/teams/PV429MG In this example, the team ID is `PV429MG`.
    - **Base parallelism**: It is recommended to use a higher value than 1 so that you can process data in parallel. In a small environment the available flinks slots is 16 and for a large environment, the maximum available slots is 32.
    - **Sampling rate**: The rate at which data is pulled from live source (in seconds). The default value is `60`.
    - **Data flow**: Set the toggle to 'On' to enable event data from PagerDuty to flow to IBM Cloud Pak for AIOps. For more information about enabling your connection, see Enabling PagerDuty connections.

      Note: When the `body.cef\_details` attribute is not present in the raw data from PagerDuty, there is no way to extract the entity and assign it to the `resource` attribute in the event input and the event cannot be associated with an application.

      You can test your connection by clicking **Test connection**.

    Note: PagerDuty mappings are the default mappings for IBM Cloud Pak for AIOps's standard fields for events.
7. Click **Done**.

You created a PagerDuty connection in your instance.

To create more connections, see Configuring data and tool connections.

For more information about working with the insights provided by your connections, see ChatOps insight management.

Enabling and disabling PagerDuty connections

If you didn't enable your data collection during creation, you can enable your connection afterward. You can also disable a previously enabled connection the same way. To enable or disable a created connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Manage connections* tab of the *Data and tool connections* page, click the **PagerDuty** connection type.
4. Click the connection that you want to enable or disable.
5. Set **Data collection** to **On** or **Off** to enable or disable log data collection. Disabling data collection for a connection does not delete the connection.

You enabled or disabled your connection. For more information about deleting a connection, see Deleting PagerDuty connections.

Editing PagerDuty connections

After you create your connection, your can edit the connection. To edit a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **PagerDuty** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the PagerDuty connections page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.
5. Edit your connection. Click **Next** to go through the connection configurationn pages. Click **Save** when you are done editing.

You edited your connection. If you did not previously enabled or disabled your application, you can enable or disable the connection directly from the interface. For more information about enabling and disabling your connection, see Enabling and disabling PagerDuty connections. For more information about deleting a connection, see Deleting PagerDuty connections.

Deleting PagerDuty connections

If you no longer need your PagerDuty connection and want to not only disable it, but delete it entirely, you can delete the connection from the console.

Note: You must disable data collection before you delete your connection. For more information about disabling data collection, see Enabling and disabling PagerDuty connections.

To delete a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **PagerDuty** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the PagerDuty connections page, click the options menu (three vertical dots) for the connection that you want to delete and click **Delete**.
5. Enter the name of the connection to confirm that you want to delete your connection. Then, click **Delete**.

Your connection is deleted.

Creating a ServiceNow connection

A ServiceNow connection collects historical and live data for change request incidents and problems. AI algorithms in IBM Cloud Pak® for AIOps use this data to calculate change risk scores and update them in the ServiceNow interface. Incidents that are triggered in IBM Cloud Pak for AIOps can also be published as incidents in ServiceNow.

In addition, it automatically generates alerts in the *Incidents and alerts* section of the IBM Cloud Pak Automation console whenever it encounters common errors that go unnoticed by the user. These errors relate to the overall health and monitoring issues of the IT environment.

The ServiceNow connection triggers the following alerts:

- Instance is hibernating
- ServiceNow credentials unauthorized
- Instance does not exist or is deleted
- IBM Cloud Pak for AIOps ServiceNow integration is not installed on ServiceNow

For more information about alerts, see About events, alerts, and incidents.

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

About this task

Similar incidents, such as from a ServiceNow connection, are called similar tickets. Similar tickets are grouped with other tickets from other connections as input to IBM Cloud Pak for AIOps AI algorithms.

If your connected ServiceNow application is version 4.1 or later, incidents update through the connection.

- When you make updates to an IBM Cloud Pak for AIOps incident state, priority, or description, the updates are reflected in ServiceNow. For example, when an incident is closed in IBM Cloud Pak for AIOps, the incident is closed in ServiceNow.
- The following limitations apply to updating incidents through the connection:
  - Incident updates are sent only from IBM Cloud Pak for AIOps to ServiceNow, and alerts are not updated directly. For example, closing an incident in ServiceNow does not close the incident in IBM Cloud Pak for AIOps.
  - Also, alerts in ServiceNow are updated only when their associated IBM Cloud Pak for AIOps incidents are updated. Updates to IBM Cloud Pak for AIOps alerts, such as an increased event count, are not reflected in their ServiceNow counterparts. ServiceNow alerts might become out of sync with IBM Cloud Pak for AIOps alerts. View current alert data in the IBM Cloud Pak for AIOps Alert Viewer. For more information, see Alert Viewer.
  - If your ServiceNow application version is older than version 4.1, errors occur.

Prerequisites

- Before you create a ServiceNow connection, you must create users and configure ServiceNow. For more information about configuring ServiceNow, see Configuring the ServiceNow App for IBM Cloud Pak for AIOps.
- Your ServiceNow application must be version 4.2.1 to connect with IBM Cloud Pak® for AIOps version 4.2.0. This connection is not supported for earlier versions of the ServiceNow application.
- You can have only one ServiceNow connection per instance.
- ServiceNow instances with self-signed certificates are not supported for connecting with IBM Cloud Pak for AIOps.

For more information about working with a ServiceNow connection, see the following sections:

- Creating a ServiceNow connection
- Enabling a ServiceNow connection
- Editing a ServiceNow connection
- Deleting a ServiceNow connection
- Troubleshooting a ServiceNow connection
- Known issue with a ServiceNow connection

For information about REST APIs and ServiceNow, see Configuring ServiceNow Observer jobs and Topology REST Observer APIs.

Creating a ServiceNow connection

To create a ServiceNow connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.
4. From the list of available connections, find and click the **ServiceNow** tile.

   Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Connect**.
6. On the *Add connection* page, define the general connection details:
   - **Name**: The display name of your connection.

- **Description**: An optional description for the connection.
- **URL**: The URL of your ServiceNow instance.

  Important: If you have any restrictive EgressNetworkPolicies in place, ensure that they are updated to allow for this outbound connection. For more information, see Postinstallation tasks.
- **User ID**: A ServiceNow user with a **`ibm.aiops`** user ID and an **`x_ibm_waiops.admin`** role.

  Important: You must create a new, unique user in ServiceNow specifically for your ServiceNow connection before you create your connection. Make sure that you are using the correct user role, the *x_ibm_waiops.admin*, for your connection. You must also make sure that the time zone set for this user matches your system time zone in ServiceNow. For more information about ServiceNow users, see Managing roles.
- **Password**: Your ServiceNow password for **`x_ibm_waiops.admin`**.
- Select the orchestration or deployment type as **Local** or **Remote**.

  Note: If you select 'Remote' you will be provided with a bootstrap command to finish the orchestration.

Click **Test Connection** to test the connection.

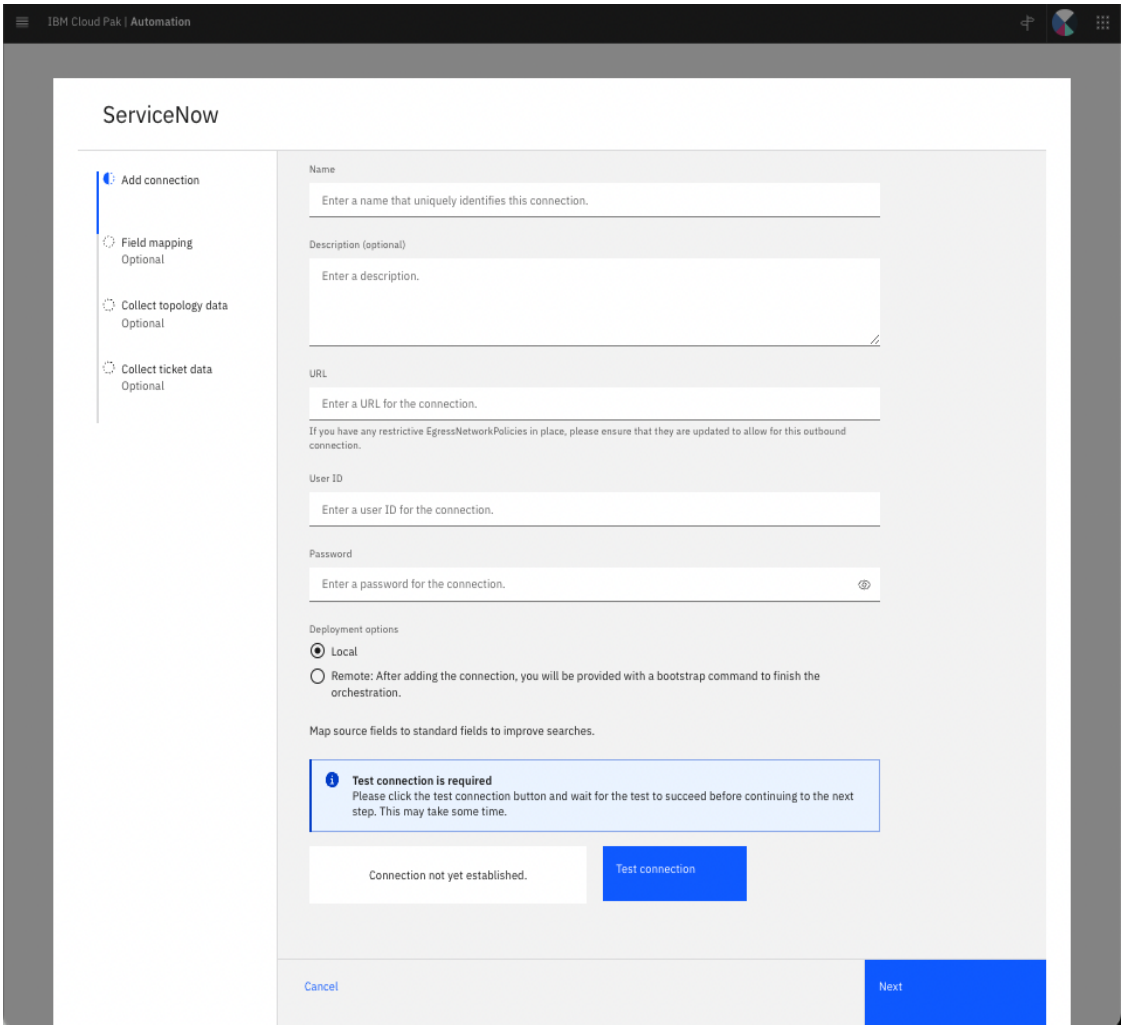


Figure. Test connection button

7. Click **Next**.
8. (Optional) Enter *Field mapping* information.

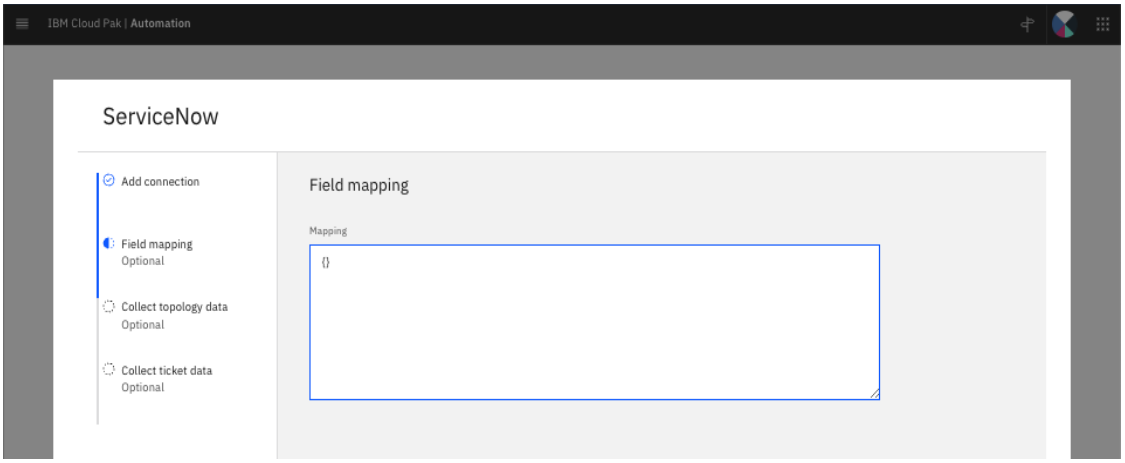


Figure. Field mapping setting

With this field, you can map custom data fields to default fields in ServiceNow. The **change_request** and **incident** tables are queried from ServiceNow. These fields are used for AI training. These tables can be modified to include additional columns.

- The following **change_request** fields are extracted:

```
"u_environment", "u_environment_id", "cmdb_ci", "short_description", "u_purpose",
"description", "backout_plan", "close_notes", "state", "close_category",
"sys_created_by", "assigned_to", "sys_domain", "category", "business_service",
"priority", "approval", "type", "contact_type", "production_system", "impact",
"reason", "justification", "start_date", "end_date", "work_start", "work_end",
"calendar_duration", "close_code","delivery_task", "implementation_plan",
"test_plan", "work_notes_list"
```

- The following **incident** fields are extracted:

```
"short_description", "description", "u_caused_by_change", "caused_by", "close_code",
"u_detection_source", "close_notes", "u_environment","cmdb_ci",
"comments_and_work_notes", "impact", "problem_id", "priority", "severity", "state",
"resolved_at", "closed_at", "opened_at", "parent", "made_sla", "watch_list",
"upon_reject","sys_updated_on", "child_incidents", "hold_reason",
"approval_history", "resolved_by", "sys_updated_by","opened_by", "user_input",
"sys_created_on", "sys_domain", "sys_created_by", "knowledge",
"order","calendar_stc", "delivery_plan", "contract", "active", "business_service",
"sys_domain_path", "rfc","time_worked", "expected_start", "business_duration",
"group_list", "work_end", "caller_id", "reopened_time", "approval_set",
"subcategory", "correlation_display", "delivery_task", "work_start",
"assignment_group", "additional_assignee_list", "business_stc", "calendar_duration",
"notify", "service_offering", "sys_class_name", "closed_by", "follow_up",
"parent_incident", "contact_type", "reopened_by", "incident_state", "urgency",
"company", "reassignment_count", "activity_due", "assigned_to", "comments",
"approval", "sla_due", "due_date", "sys_mod_count", "reopen_count", "sys_tags",
"escalation", "upon_approval", "correlation_id", "location", "category",
"work_notes_list"
```

Example: If you add a custom column (named **my_custom_field1_name**) into the table, the prefix **u_** is added so the field must be queried as **u_my_custom_field1_name**. If this field contains useful description information for training, you might want the existing **description** field to be updated for both change requests and incidents to use this data. To support this mapping, you need to add the following JSON to the *Field mapping*:

```
{
  "codec": "servicenow",
  "description": "u_my_custom_field1_name"
}
```

With this mapping, the **description** value is replaced with the value from **u_my_custom_field1_name**. For example, assume the default **description** field has the value **Server down for 2 hours due to an upgrade** and the **u_my_custom_field1_name** field had the value **Impact: server down, Duration: 2 hours, Cause: server upgrade**. With the above mapping in place, the **description** now has the value **Impact: server down, Duration: 2 hours, Cause: server upgrade**. This **description** value is also now used for AI

training. If the mapping is not in place, the **`u_my_custom_field1_name`** value is not used for AI training as it is not examined throught the connection.

Adding the field mapping can give you flexibility to add your own custom fields into AI training. Also, if an existing field should not be used for training, you can add a new custom column with a default value to use for training instead of the existing field value.

Notes:

- The field mapping is applied to both historic and live data. Once the mapping is set in the connection and training is completed, the mapping should not be changed. Otherwise, if new change requests or incidents are queried, the mapping is not applied properly, which can cause results to be less accurate.
- Currently, the field mapping applies to all tables. If a custom field does not exist in a table, a workaround is to create that custom field with default values to ensure that the mapping does not fail. Otherwise, the data is not be retrieved properly for that table.

9. (Optional) Enter the *Collect topology data* and *Set advanced options* information:

    - **Collect inventory and topology data**: Set to **On** for inventory and topology data collection. You can also set advanced options.
    - **Schedule request**: Set to **On** and this allows you set the **Job Schedule** (optional)
        - **Start date**: The start date of when the job is to run.
        - **Time**: The time to run the observer job.
        - **Time interval (period)**: How frequently to run the job (either by hour, or by minute).
        - **Interval**: The duration of time between runs based on the *Time interval (period)*. For example, if you wanted the job to run every 2 hours, set the interval to **`2`** and the *Time interval (period)* to *hours*. Enter **`0`** for the *Interval* to run the observer job one time (and manually through the interface otherwise).
    - **Maximum number of records to retrieve per API call (100-10000)**: Number of records to retrieve per API call (100 - 10000). The default value is 10000.
    - **Resource type mapping**: Map ServiceNow resource types to Topology Manager resource types in the form c_type_1:asm_type_1,ci_type_2:asm_type_2. Topology Manager resource types should not include spaces.
    - **Discoverable CI types**: Specify ServiceNow Configuration Items (CI) types to discover in the form sys_class_name_1,sys_class_name_2. To discover all CI types, leave the field blank.
    - **Connection timeout**: The amount of time before a connection times out in milliseconds. The default value is 5000 milliseconds.
    - **Read timeout**: The amount of time before a read operation times out in milliseconds. The default value is 5000 milliseconds.
    - **Proxy Host**: The proxy host for your ServiceNow instance.
    - **Proxy Port**: The proxy port for your ServiceNow instance. The default value is **`8080`**.

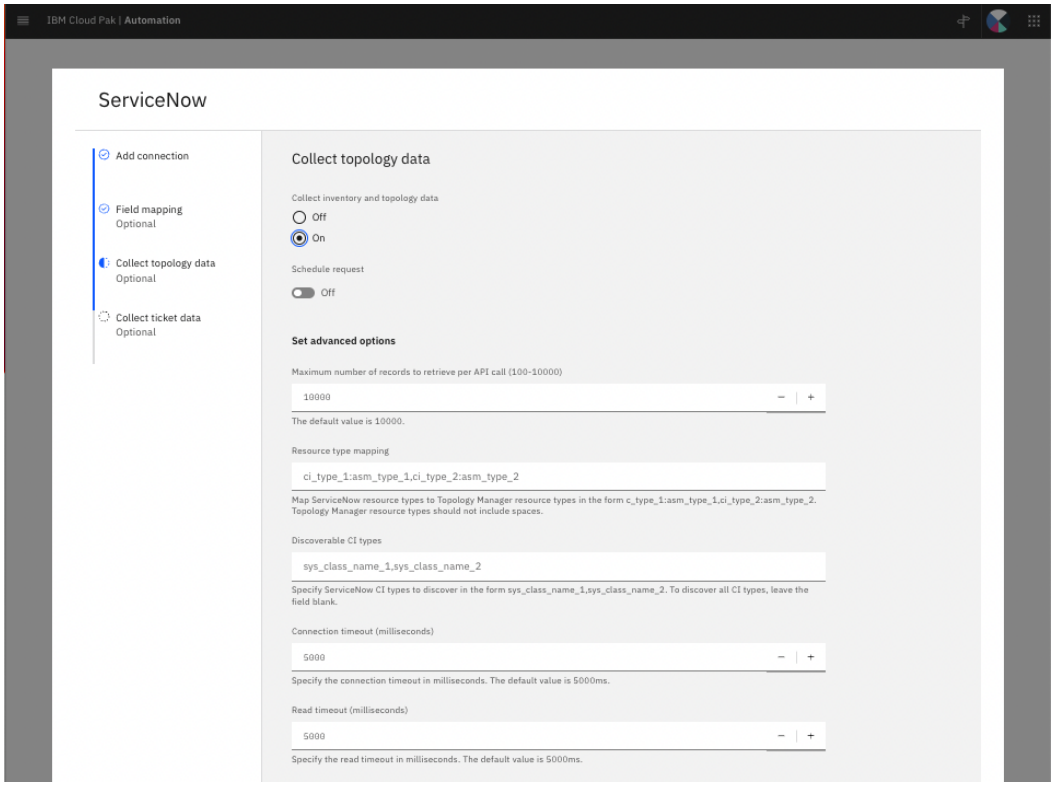


Figure. Topology data collection settings

For information about ServiceNow jobs, see Configuring ServiceNow Observer jobs.

10. Click **Next**.
11. (Optional) Enter the *Collect ticket data* information. Set the **Data collection** toggle set to **On** to collect data. Then, select how you want to collect data:

    - **Live data for continuous ticket data collection**: To generate incidents and change risk assessments (after a model is trained).

      Note: Before you select this option, you must first set **Mode** to **Historical data for ticket data collection** to collect a minimum amount of data, and then run AI training on that data. For more information, see Planning data loading and training.

    - **Live data for initial ticket data collection**: A single set of training data used to define your AI model. Data collection takes place over a specified time period that starts when you create your connection.

      Note: If you select this option, you must disable your connection when you collect enough data for training. If you do not disable the connection, it continues to collect data. For more information about AI model training, including minimum and ideal data quantities, see Configuring AI training. For more information about disabling and enabling a connection, see Enabling and disabling a ServiceNow connection.

    - **Historical data for ticket data collection**: To train a change risk model (required to enable change risk assessment functions). A single set of training data used to define your AI model. You complete the 'Start' and 'End' date fields.

    - **Ticket types**: Specify the types of ServiceNow tickets that you want to pull in to the system. By default, change request, incident, and problem tickets are included and data is queried from those ServiceNow tables. You can edit this list to remove any types you don't want to sample. For change requests, the default sampling rate is 10-second intervals, to support the real-time requirements for change risk.

    - **Sampling Rate**: You must enter a sampling rate between 1 and 60 minutes. The default value is `1`.

Note: If you selected *Historical data for ticket data collection*, you also need to enter a *Start* and *End* date.

Important: Different types of AI models have different requirements to properly train a model. Make sure that your settings satisfy minimum data requirements. For more information about how much data you need to train different AI models, see

Configuring AI training.

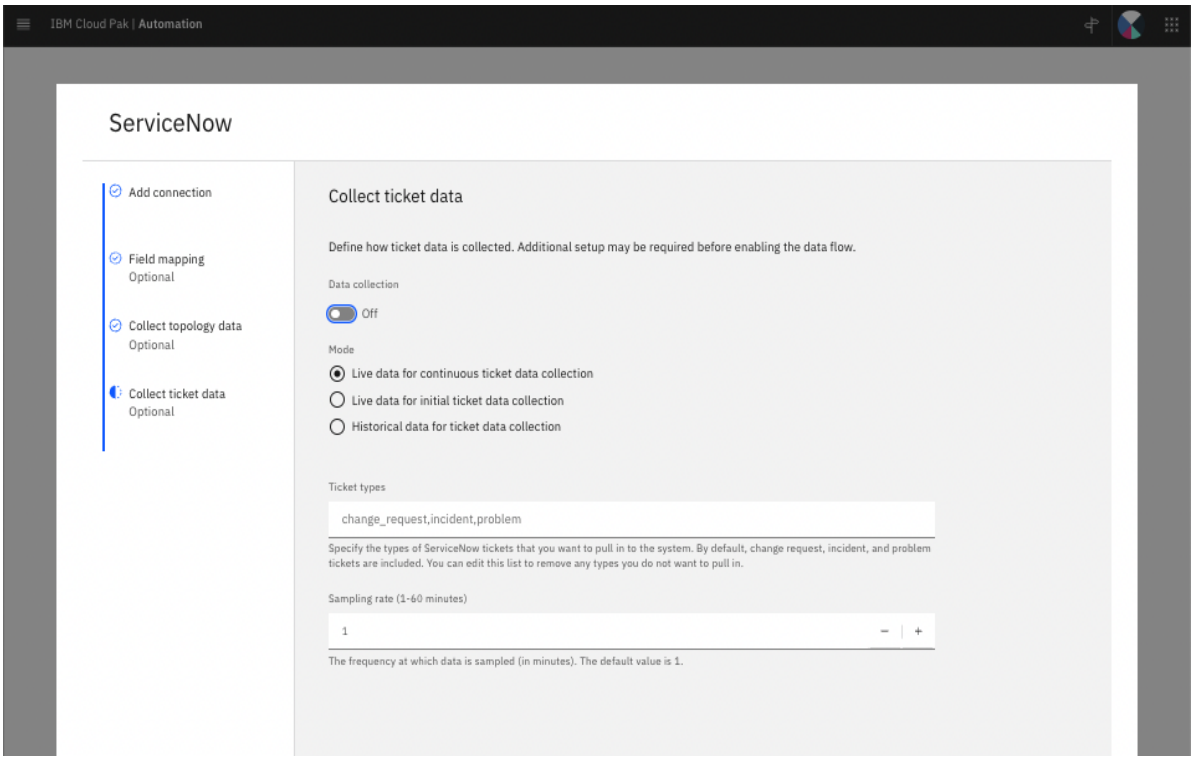


Figure. Collect ticket data settings

12. Click **Done**.

You have created a *ServiceNow* connection in your instance.

After the connection creation is complete, and if you selected the *Remote* option for connector deployment, you can download the *bootstrap script* by selecting the created connection, and clicking **download** from the options menu (three vertical dots). If a script is regenerated, it invalidates the credentials of previously downloaded scripts for that connection.

Note: You can either copy the script or download as an sh file to run it on a remote cluster.

Prerequisites before running the bootstrap script on remote cluster:

- Podman needs to be installed on the remote cluster.
- Log in to the *Image registry* for the script to pull in the image that uses the Podman login command.

After you create your connection, you must enable the data collection to connect your connection with the AI of IBM Cloud Pak for AIOps. For more information about enabling your connection, see Enabling a ServiceNow connection. To verify that your job completed successfully, view the schedule status of your **ServiceNow** connection in the *Schedule connections* tab.

To create more connections (such as a ChatOps connection), see Configuring data and tool connections.

For more information about working with the insights provided by your connections, see ChatOps insight management. For more information about change risk from ServiceNow, see Change risk.

Training a Change Risk model

Note: Make sure ServiceNow Integration is first set to *Historical* mode.

1. Go to IBM Cloud Pak Automation console.
2. Click **AI Model Management**.
3. Find the Change risk algorithm under the *Training* tab.
4. Click **Set up training**.

5. Verify your ServiceNow integration is listed under *Data & tool connections*.

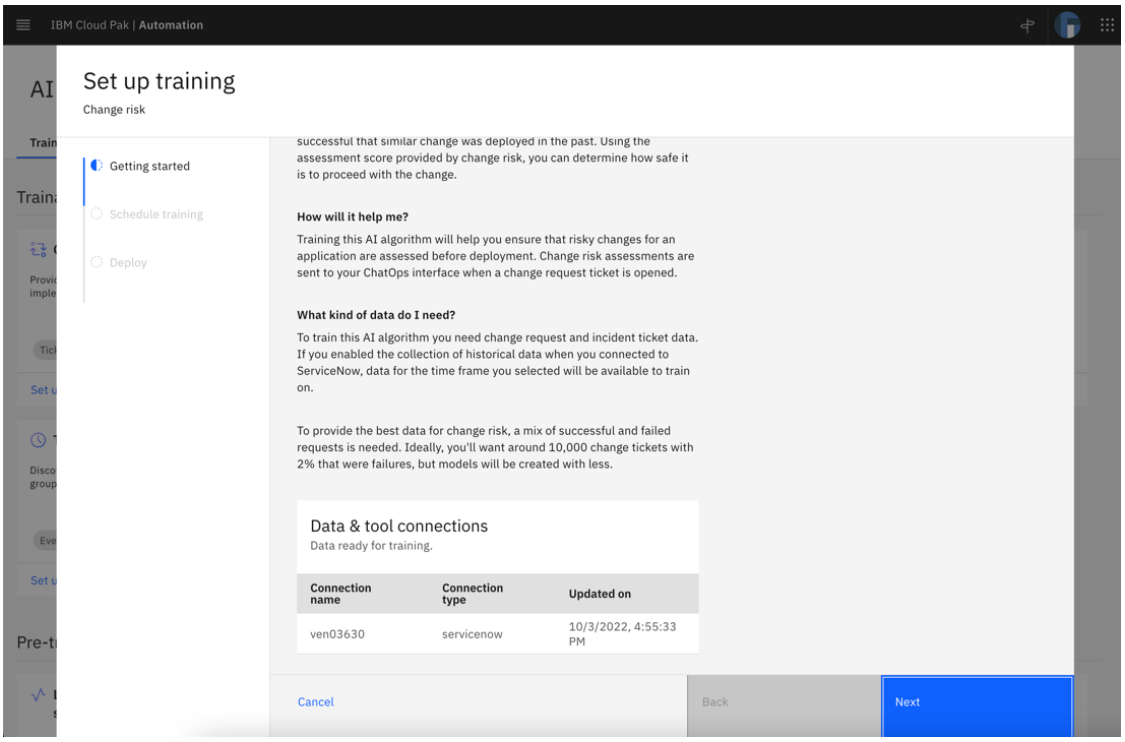


Figure. Set up training page

6. Click **Next** for **Schedule training**. No need to schedule the training. It starts training in a few moments.
7. Click **Next** for **Deploy**. Set to deploy model *On completion*.
8. Click **Done**.
9. View details on the Change risk algorithm.

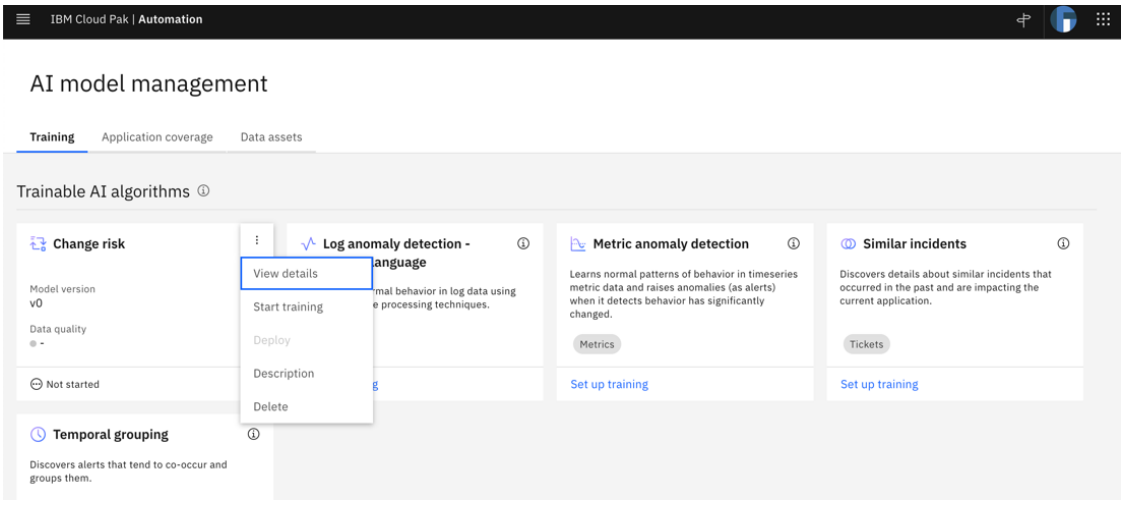


Figure. View details menu option

10. Click **Start training**. IBM Cloud Pak® for AIOps will first check data quality before training.

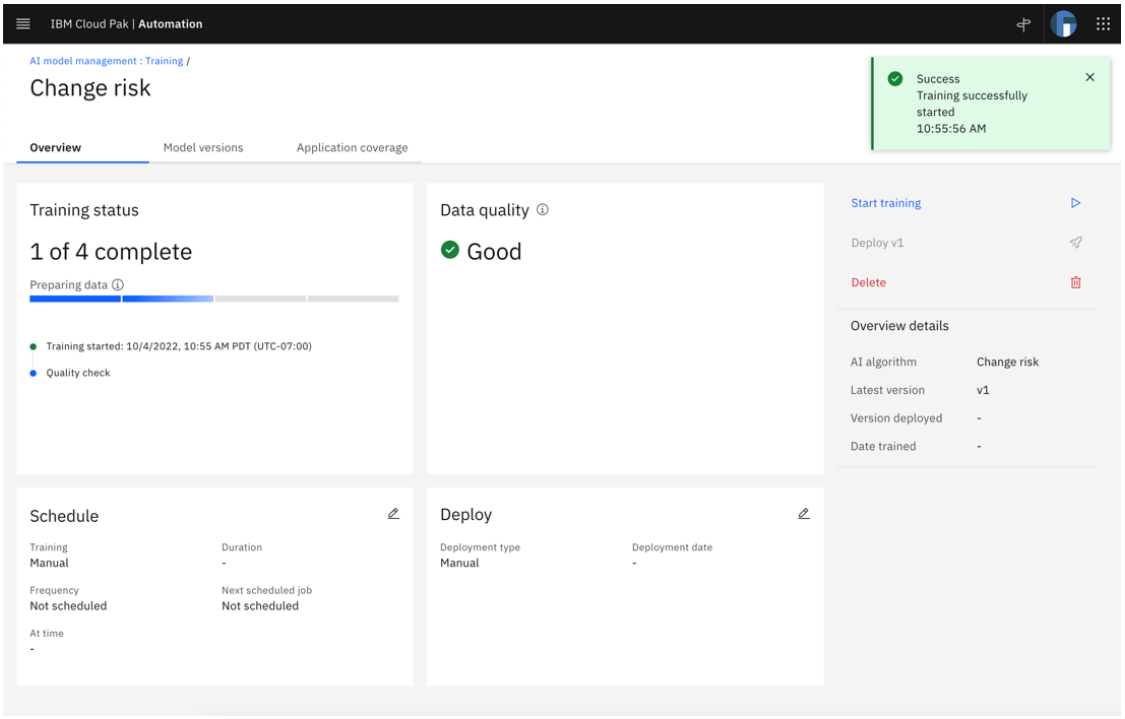


Figure. Data quality check

11. Deploy model to enable change risk assessment functionality. Remember to change ServiceNow Integration to Live data mode

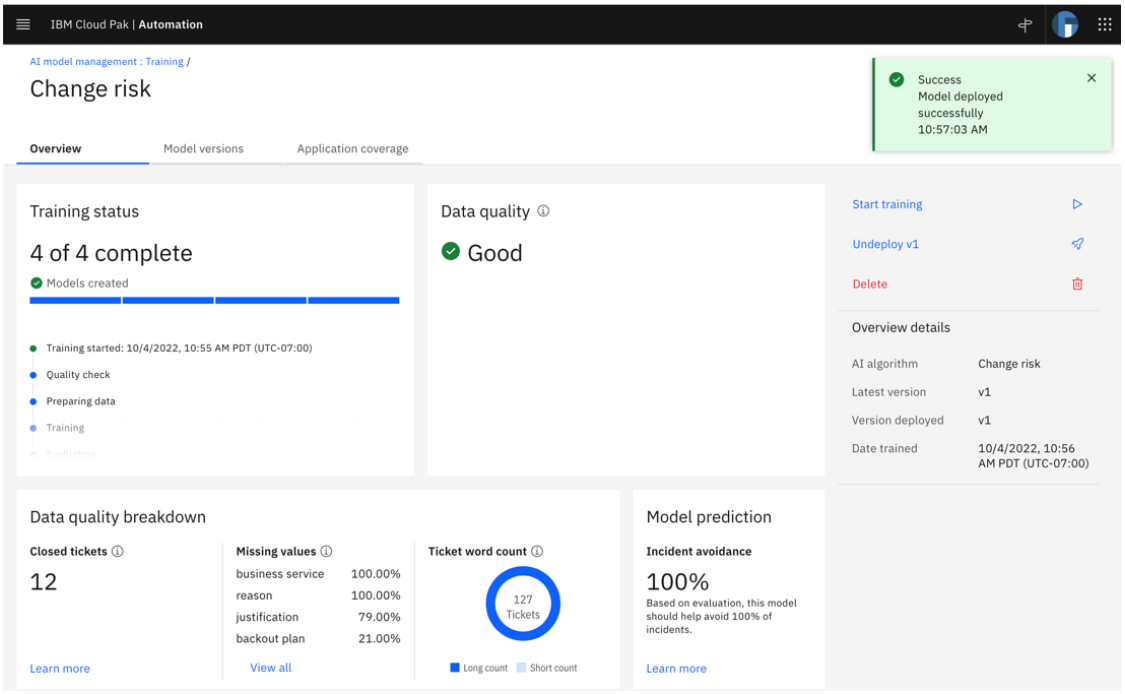


Figure. Model deployed successfully

Enabling and disabling a ServiceNow connection

If you didn't enable your data collection during creation, you can enable your connection afterward. You can also disable a previously enabled connection the same way. If you selected *Live data for initial AI training* when you created your connection, you must disable the connection before AI model training. To enable or disable a created connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.

2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Manage connections* tab of the *Data and tool connections* page, click the **ServiceNow** connection type.
4. Click the connection that you want to enable or disable.
5. Go to the *Collect ticket data* section. Set **Data collection** to **On** or **Off** to enable or disable data collection. Disabling data collection for a connection does not delete the connection.

You enabled or disabled your connection. For more information about deleting a connection, see Deleting a ServiceNow connection.

Editing a ServiceNow connection

After you create your connection, your can edit the connection. For example, if you specified *Historical data for initial AI training* but you now want your connection to sample live data for continuous monitoring, you can edit it. To edit a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **ServiceNow** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the ServiceNow connections page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.
5. Edit your connection. Click **Next** to go through the connection configurationn pages. Click **Save** when you are done editing.

You edited your connection. If you did not enable ticket data collection when creating your connection, you can enable the connection directly from the interface. For more information about enabling and disabling your connection, see Enabling and disabling a ServiceNow connection. For more information about deleting a connection, see Deleting a ServiceNow connection.

Note: After you upgrade, you might find that the **Next run** field in the **Scheduled connections** tab is showing the last run time of the scheduled job, instead of the next run time. If this happens, the next run time should be regarded as the last run time.

Deleting a ServiceNow connection

If you no longer need your ServiceNow connection and want to not only disable it, but delete it entirely, you can delete the connection from the console.

Note: You must disable ticket data collection before you delete your ServiceNow connection. For more information about disabling ticket data collection, see Enabling and disabling a ServiceNow connection.

To delete a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **ServiceNow** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the ServiceNow connections page, click the options menu (three vertical dots) for the connection that you want to delete and click **Delete**.
5. Enter the name of the connection to confirm that you want to delete your connection. Then, click **Delete**.

Your connection is deleted.

Note: Alerts that are generated from the ServiceNow connector are automatically cleared when the connection is deleted.

Troubleshooting a ServiceNow connection

If the ServiceNow change request, incident, or problem contains a large amount of text, such as work notes with close to 150,000 characters, the ticket is dropped and a warning is logged in the pod log. Dropping the ticket affects change risk for that ticket, as it will not occur or that ticket will not be used for similar ticket detection.

Known issue with a ServiceNow connection

If there is an active ServiceNow connection with data collection enabled and the ServiceNow credentials change, the ServiceNow user account can get locked out. ServiceNow has an automatic login locking script that is called "SNC User Lockout Check", which locks users out after more than five failed attempts (including any failed API calls).

If you check the *Incidents and alerts* page, you see also an alert saying "ServiceNow instance authentication failed".

When this problem occurs, unlock the user in ServiceNow. Then, change the password in the ServiceNow connection and save. When authentication fails in the ServiceNow connector, there is a 1-minute wait time before you can access it, to prevent a lockout from occurring quickly.

Mapping similar ticket data from ServiceNow

If you need to use historical incidents from ServiceNow as similar ticket data, you need to map the data to an expected format and insert the data into Elasticsearch.

Historical incidents are required to first be inserted into Elasticsearch for similar ticket data. The data format consists of one incident record per line, in JSON format, with the original raw data. Raw data must be normalized to the IBM Cloud Pak for AIOps target format. This normalization includes mapping fields, converting date-time formatting, and more. Incidents are then inserted in batch. When the insertion is done, IBM Cloud Pak for AIOps tries a simple search to make sure everything runs as expected.

For more information about training similar tickets models in IBM Cloud Pak for AIOps, see Setting up training for similar tickets.

Raw similar ticket data

IBM Cloud Pak for AIOps ingests incidents from standard or custom data sources, for the incidents to be analyzed for similarities and action recommendations. Offline historical raw incidents are first uploaded to the Elastic database to extract indexes and set the stage for finding incidents that are similar to incoming live incident data.

To maintain the structure of the incident data in the Elastic database, you must transform raw incident data into a normalized format. That normalized data can then be used in the training pipeline. For example, the following raw JSON incident data comes from ServiceNow:

```
{
    "number": "INC0010066",
    "short_description": "Containers being killed in ts-payment-mongo service. Users may
notice issues during payment",
    "description": "",
    "opened_at": "2020-01-08T22:48:23Z",
    "sys_created_on": "2020-01-08T22:48:23Z",
    "sys_updated_on": "2020-01-08T22:48:23Z",
    "resolved_at": "2019-12-11T07:36:00Z",
    "closed_at": "2019-12-11T07:36:00Z",
    "business_duration": "",
    "severity": 3,
    "priority": 5,
    "impact": 3,
    "state": "closed",
    "comments_and_work_notes": "",
    "close_notes": "This issue has been resolved by the DBA team, which SQL queries were
updated to reboot proxy agent to prevent the monitoring notification system from failing.",
    "parent_incident": ""
}
```

In this example, you can extract data from ServiceNow, then map that raw incident data to the IBM Cloud Pak for AIOps normalized output schema. The source and format of your exported data doesn't matter. That data must map to the IBM Cloud Pak for AIOps normalized output data schema for training.

For more information about exporting data from ServiceNow, see Exporting data.

For more information about managing incident data in ServiceNow, see Incident management.

Normalized similar ticket data

The following sample illustrates the result of mapping data from an external source for use with IBM Cloud Pak for AIOps:

```
{
  "application_group_id": "1",
  "application_id": "1",
  "timestamp": 1570221600,
  "utc_timestamp": "2019-10-04T20:40:00.038Z",
  "type": "incident",
  "incident": {
    "incident_id": "fed6b7bf-f741-49d1-ab36-5d04e9a5c128",
    "title": "Users are not able to access Discovery and Conversation in EU",
    "description": "Unable to connect to email using my laptop.",
    "created_at": "2019-09-11T06:39:48.000Z",
    "updated_at": "2019-09-11T07:57:58.000Z",
    "resolved_at": "2019-09-11T07:57:58.000Z",
    "closed_at": "2019-09-11T07:57:58.000Z",
    "started_at": "2019-09-11T07:57:58.000Z",
    "business_duration_ms": 15347,
    "severity": 1,
    "priority": 2,
    "impact": 3,
    "state": "new",
    "source": {
      "source_name": "ServiceNow",
      "source_url":"https://ibm.servicenow.com/INC0000060",
      "source_incident_id": "INC0000060",
      "source_application_id": "tokenizer124"
    },
    "comments": [
      {
        "comment_text":"Rebooted it."
      }
    ],
    "related_incidents":[
      {
        "source_incident_id":"INC0000059",
        "relationship":"parent"
      }
    ],
    "resolution": {
      "rca_id": "RCA00013547",
      "resolution_summary": "The problem was resolved by rebooting the laptop."
    },
    "features": [
    ]
  },
  "meta_features": []
}
```

Description of similar ticket normalized attributes

Required attributes expected as input to the pipeline

Attribute	Description
`title`	Short description or title of the incident.
`description`	Long description of the incident.

Attribute	Description
`timestamp/utc_timestamp`	Epoch timestamp of the event in the log entry and the `utc_timestamp` formatted as `yyyy-mm-ddTHH:MM:SSZ`.
`created_at`	Coordinated Universal Time at which the incident was created formatted as `yyyy-mm-ddTHH:MM:SSZ`.
`updated_at`	Coordinated Universal Time at which the incident was last updated formatted as `yyyy-mm-ddTHH:MM:SSZ`.
`resolved_at`	Coordinated Universal Time at which the incident was resolved formatted as `yyyy-mm-ddTHH:MM:SSZ`.
`closed_at`	Coordinated Universal Time at which the incident was closed formatted as `yyyy-mm-ddTHH:MM:SSZ`.
`started_at`	Coordinated Universal Time at which the incident started formatted as `yyyy-mm-ddTHH:MM:SSZ`.
`severity`	Severity of the issue (integer)
`priority`	Priority of the issue (integer)
`impact`	Impact of the issue (integer).
`source_name`	Data source in which the incident is stored (for example, ServiceNow).
`source_url`	URL of the incident in the source.
`source_incident_id`	`incident_id` from the source (for example, `NUMBER` in ServiceNow).
`source_application_id`	Application from which the incident came (for example, the configuration item in ServiceNow).
`comments`	Comments or closed notes that talk about the list of actions that are taken, or comments that are added by the user who is attending to the incident.
`related_incidents`	Any related incidents, such as parent incidents, child incidents, or CIE documentation that is attached with the incident.
`resolution`	Any RCA ID or resolution that is provided for the incident.

Optional attributes that might add value

Attribute	Description
`business_duration`	Duration for which the incident was alive without resolution.

Static attributes

Attribute	Description
`application_group_id`	The `application_group_id` to support multitenancy. This value is provided as part of the initial configuration.
`application_id`	Application that IBM Cloud Pak for AIOps is monitoring. This value is provided as part of the initial configuration.
`type`	This value is hardcoded to the incident.
`timestamp`	Epoch time of the incident normalization, in seconds.
`utc_timestamp`	Normalization time in the Coordinated Universal Time time zone.
`incident_id`	The unique UUID.
`state`	Current state of the incident.

Attributes analyzed and populated in the pipeline

Attribute	Description
`features`	Placeholder for any features.
`meta_features`	Placeholder for service analysis.

Creating Splunk connections

A Splunk connection collects log data and different types of metrics, including live and historical data, and sends it to the gRPC server as Cloud events at specified intervals.

Log data is used to establish a baseline of normal behavior and detect anomalies. Metric data helps to monitor your environment and process alerts. Anomalies can be correlated with alerts, and published to your ChatOps interface, to help you

determine the cause and resolution of a problem. Splunk v8.1 is the version that is supported for the Splunk Connector.

To activate the connection, you must have an endpoint and a user ID and password credentials or a token. You can also configure settings for performance by providing log or metric data parameters. The metric anomaly detection and Log anomaly detection AI algorithms analyze the Splunk data.

For more information about working with Splunk connections, see the following sections:

- Creating Splunk connections
- Enabling Splunk connections
- Editing Splunk connections
- Deleting Splunk connections

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating Splunk connections

About this task

Before you create a Splunk connection, consider the following details:

- **Load**: To prevent the connection from placing an inordinate load on your data source and potentially impacting your logging operations, the connection connects to only one API with a default data frequency of 60 seconds. This frequency is controlled by the **Sampling rate** setting that is configured when you create the connection.
- **Access**: Custom data sources are cloud-based REST APIs. Access is configured by using the authentication methods that are specified in the **Authentication type** setting when you create the connection.
- **Data volume**: Data volume depends on the application, and is not a set value. Therefore, it does not appear in the settings.

Procedure

To create a Splunk connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.
4. From the list of available connections, find and click the **Splunk** tile.

   Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Connect**.
6. On the *Add connection* page, define the general connection details:
   - **Name**: Set a display name for your connection. Use a name that uniquely identifies this connection.
   - **Description (optional)**: Provide a description for the connection.
   - **Splunk service URL**: Enter the URL for your Splunk instance.

     Note: If you have any restrictive EgressNetworkPolicies in place, ensure that they are updated to allow for this connection.
   - **Authentication type**: Select one of the following options:
     - **User ID/Password**: The Splunk instance has a user ID and password as authentication. You must enter both in the connection configuration that appears.
     - **Token**: The Splunk instance is authenticated with a temporary token field that appears.
   - **Splunk dashboard URL**: Enter the dashboard URL for your Splunk instance.

- **Certificate (optional)**: Provide a certificate to verify the SSL/TLS connection to the REST service.
- **Deployment options**: Select one of the two options, **Local** or **Remote**. Use **Local** to deploy in the same cluster as IBM Cloud Pak Automation console. Select **Remote** to deploy anywhere, such as on a different network region, on SaaS, or on a remote on-premises system (VM, cluster, container).

  Note: Remote deployment only supports metric connectors. If you need to use the log connector, you must select **Local**. If you select **Remote**, after adding the connection, you are provided with a bootstrap command to finish the deployment.

7. Click **Test connection** (required). The test can take some time to complete.
8. Click **Next**.
9. On the *Collect log data* page, define how you want to collect log data for use in AI training:
   - **Data collection**: Use the toggle to turn on or off data collection.
     - Set the toggle to **Off** (default) if you do not want to collect log data. With this setting, no data is collected through the connection, which results in no anomalies being detected as no AI model is trained. You must turn the toggle to **On** to begin any data flow through the connection.
     - Set the toggle to **On** to turn on data collection. Once you turn on the data collection.

       Note: Log data collection is not supported for a remote deployment. If you want to collect log data, you must deploy a local connection.
   - **Mode**: Select how you want data to be collected through the connection for AI training:
     - **Live data for continuous AI training and anomaly detection**: This option enables continuous live data collection through the connection to train AI models and analyze data for anomalous behavior.

       Important: Before you use this option, you must first set **Mode** to **Historical data for initial AI training** and collect a minimum amount of data. Then, you need to run AI training on that data. For more information, see Planning data loading and training.
     - **Live data for initial AI training**: This option collects a single set of training data to define your AI model. Data collection takes place over a specified time period that starts when you create your connection.

       Note: If you select this option, you must disable your connection when you collect enough data for training. If you do not disable the connection, it continues to collect data. For more information about AI model training, including the minimum and ideal data quantities, see Configuring AI training. For more information about disabling and enabling the connection, see Enabling and disabling Splunk connections.
     - **Historical data for initial AI training**: This option collects a single set of training data to define your AI model. Historical data is harvested from existing logs in your connection over a specified time period in the past. You must specify the start and end dates for this data collection, along with the parallelism of your source data.
       - **Start date**: Select a start date from the calendar. The start date must not exceed 31 days from the present as the maximum time period for historical data collection is 31 days. The recommended time period is two weeks.
       - **End date**: Select the end date from the calendar. The recommended time period is two weeks.
       - **Source parallelism (1-50)**: Select a value to specify the number of requests that can run in parallel to collect data from the source. Generally, you can set the value to equal the number of days of datat that you want to collect. When you are setting this value, consider the number of requests that are allowed by the source in a minute. For example, if only 1-2 requests are allowed, set the value to be low.

   Important: Keep in mind the following considerations when you select your data collection type:
   - Anomaly detection for your connection occurs if you select *Live data for continuous AI training and anomaly detection*.

- Different types of AI models have different requirements to properly train a model. Make sure that your settings satisfy minimum data requirements. For more information about how much data you need to train different AI models, see Configuring AI training.

- **Mapping**: Enter the field-mapping information.

  You can improve search performance by mapping your implementation fields to the standard IBM Cloud Pak for AIOps fields. For more information about this mapping, including how to clean data for use in IBM Cloud Pak for AIOps, see, Mapping data from incoming sources.

  The following code shows the supported data schema for the connection:

  ```
  {
    "rolling_time": 10,
    "instance_id_field": "sourcetype",
    "log_entity_types": "source",
    "message_field": "_raw",
    "timestamp_field": "_time",
    "resource_id": "sourcetype"
  }
  ```

- **Filters (optional)**: Enter source, app, or log-level filters to apply to the event data. For more information, see Search Tutorial in the Splunk documentation.

- **Base parallelism (1-50)**: Select a value to specify the number of Flink jobs that can run in parallel. These jobs run to process and normalize the collected data. The default value is 1. However, it is recommended to use a higher value than 1 so that you can process data in parallel. This value cannot exceed the total available free Flink slots. In a small environment, the available flinks slots are 16, while in a large environment, the maximum available slots are 32. If you are collecting historical data with this connection, you can set this value to be equal to the **source** parallelism.

  Note: To improve data throughput, you can increase the base parallelism value incrementally. For more information about maximum base parallelism for starter and production deployments, see Improving data streaming performance for log anomaly detection.

- **Sampling rate (1-150 seconds)**: Select a rate at which data is sampled from live sources (in seconds). The default value is `60`.

- **JSON processing option**: Select a JSON processing option from the following options:

  - **None**: The default option. The JSON is not processed or modified.
  - **Flatten**: This option flattens the JSON object by removing the opening and closing braces.
  - **Filter**: This option extracts the JSON object and replaces it with an empty string.

  For more information about the options, see Managing embedded JSON.

10. Click **Next**.

11. On the *Collect metric data* page, define how you want to collect metric data for monitoring.

    - **Enable Data collection**: Set the toggle to on to enable the data collection. If data flow is disabled, a new connection is created, but metrics are not collected.

    - **Select an index to see all associated source types**: Pick a splunk index from the list to see its related source types.

    - **Select a source type to see all associated metrics**: Choose a source type from the list to see its related metrics in the filter table.

    - **Search all metrics for the current selection**: Search through all the metrics that are associated with the specified index and source type and select the wanted metrics from the table.

    - **Hosts**: Select the host server.

    - **Aggregation interval (minutes)**: Enter a value in minutes. The value represents the frequency at which data is collected. The default value is 5 minutes.

Complete the settings to use either historical or live data collection:

- To use only live data collection, complete the following settings:

  Select the checkbox **Only collect live data once the connection is established**.

  Note: If you choose to collect only live data, you cannot collect historical data after live data collection starts.

- To use historical data collection, complete the following settings:

  - Do not select the checkbox **Only collect live data once the connection is established**.
  - Set the **Start date**. Select a start date from the calendar and enter the time in hh:mm (hours and minutes) format. Select the time zone from the dropdown list, for example (Localtime or UTC).

    Note: The start date must not exceed 31 days from the present as the maximum time period for historical data collection is 31 days. The recommended time period is two weeks.

  - (Optional) Set an end date for the data collection. If you do not specify the end date, then live data collection follows the historical data collection. If you do not want to set an end date, click **Remove end date**.

    To add the end date, click **Add end date**. For the **End date**, select an end date from the calendar and enter the time in hh:mm format. The end date uses the same time zone as the start date.

12. Click **Done**. You created a Splunk connection in your instance.

After you create your connection, enable data collection to use your connection with the AI of IBM Cloud Pak for AIOps. For more information, see Enabling and disabling data collection.

For more information about performance considerations for metric data collection, see Performance considerations for metric data collection.

To create more connections (such as a ChatOps connection), see Configuring data and tool connections.

For more information about working with the insights provided by your connections, see ChatOps insight management.

Enabling and disabling Splunk connections

If you didn't enable your data collection during creation, you can enable your connection afterward. You can also disable a previously enabled connection the same way. If you selected *Live data for initial AI training* when you created your connection, you must disable the connection before AI model training. To enable or disable a created connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Manage connections* tab of the *Data and tool connections* page, click the **Splunk** connection type.
4. Click the connection that you want to enable or disable.
5. Go to the *AI training and log data* section, and set **Data collection** to **On** or **Off** to enable or disable log data collection.

   Note: Log data collection is not supported for a remote deployment. If you want to collect log data, you must deploy a local connection.

6. Go to the *Collect metric data* section, and set **Enable Data collection** to on or off to enable or disable metric data collection.

Note: Disabling data collection for a connection does not delete the connection.

You enabled or disabled your connection. For more information about deleting a connection, see Deleting Splunk connections.

Editing Splunk connections

After you create your connection, your can edit the connection. For example, if you specified *Historical data for initial AI training* but now want your connection to pull in live data for continuous monitoring, you can edit it. To edit a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Splunk** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the Splunk connections page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.
5. Edit your connection. Click **Next** to go through the connection configurationn pages. Click **Save** when you are done editing.

Your connection is now edited. If your application was not previously enabled or disabled, you can enable or disable the connection directly from the interface. For more information about enabling and disabling your connection, see Enabling and disabling Splunk connections. For more information about deleting a connection, see Deleting Splunk connections.

Deleting Splunk connections

If you no longer need your Splunk connection and want to not only disable it, but delete it entirely, you can delete the connection from the console.

Note: You must disable log data collection before you delete your connection. For more information about disabling data collection, see Enabling and disabling Splunk connections.

To delete a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Splunk** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the Splunk connections page, click the options menu (three vertical dots) for the connection that you want to delete and click **Delete**.
5. Enter the name of the connection to confirm that you want to delete your connection. Then, click **Delete**.

Your connection is deleted.

Creating an SSH connection

An SSH connection provides a connection to the system where scripts and command can be run. The connection is agentless and connects directly to the target machine. It authenticates by using SSH public key authentication.

The SSH provider works for back-end systems (targets) running UNIX or Windows. For UNIX, the user running the automation must have sufficient rights to run these features.

- `bash` - A shell that is used to wrap and run the specified commands or script.
- `mktemp` - A utility that is used to create a temporary file, which is required for the script execution with this automation provider to work.
- `openssl` - A utility that is used on the target system to decrypt the transferred commands or script.

Note: You can only create one SSH connection. Also, you must have an account with administrator role to create, edit, view, or delete an SSH connection.

Before configuring you need to provide the following information:

- General connection information

If you are using a jump server (optional), you must configure it. Depending on your environment, you might require a jump server to access your target endpoints. A jump server is an SSH endpoint that is used to connect to the nested SSH endpoints. This is a common approach that is used to communicate between different network zones. To use a jump server with RBA, it must have an SSH server that is running and the `nc` command must be available. This is used to connect to nested SSH target endpoints.

Jump server configuration information:

- **Host name/IP Address:** The hostname or IP address of the jump server.
- **Port Number:** The SSH port of the jump server.
- **Username:** The username for authentication on the jump server.
- **Password:** The password for authentication on the jump server.

Any connections to SSH target endpoints use the specified jump server.

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Creating an SSH connection

To create an SSH connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.
4. From the list of available connections, find and click the **SSH** tile.

   Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Connect**.
6. On your target machine, register the default public key to enable access to the target endpoints through SSH for all users.

   Configuring SSH public key authentication for the UNIX root user

   The displayed public key must be added to all target machines that you plan to run scripts on through the SSH Provider. This key enables any RBA user to run script automations on the given target endpoint. The key must be added to the authorized_keys file that is usually found in the /root/.ssh/authorized_keys folder.

   Configuring SSH public key authentication for a specific UNIX user

   If you want to enforce that only a specific UNIX user can run the script on this target endpoint you should copy the key to the authorized_keys file in the home directory of the specific user, for example /home/john/.ssh/authorized_keys.

   You can regenerate the public key by clicking the **refresh** button in the upper right corner of the public key.

   Note: Regenerating the public key deletes the old key pair. If you choose to regenerate the key pair, you must exchange the public key in each target machine that you plan to access through the SSH Provider.
7. On the *Manage SSH Keys* tab that opens, you will see the SSH field default public key for SSH. You need to click the **Copy SSH key** symbol to copy this key to each target system, and append it to the authorized_keys file, for example ~/.ssh/authorized_keys.

   An example of an SSH key is:

   ```
   ssh-rsa
   AAAAB3NzaC1yc2EAAAADAQABAAABAQDuZkPxIyYH9NrfLEBkIGdwc6frF7WgR9vKZavE97GlTfAAZVhqoTsXO2jLo
   28sNC7+8wkOYnFEdfBff9tcQPx/lc3d1df35/hJIT0a3jeHxw8YrU3/y6QIzynVvSgcQfKzB33wdN7n8xC5ZPWKXE
   WM7FbP58kOdzHw7f8fbBIKlPRc9SOUrC0JGvndVvpHOU7x8S3q9EJlD2nKaozA6yu2mcH38CLTNCBRRwbPZ+rxBxW
   dvJ4mMWvWtJe4lt50W2zAGCIscLKLbyyMGp/DCcJFsMhkOetBDuxAfL1ZkO7rXPT5vK5Fp6549OPDXjqfHKEJ+9WA
   SZD2ui1qmCdeQpUN
   ```

Note: Every time that you generate a new key that you must register the new key again on all target systems.

8. Click **Next**.
9. The *Define a jumpserver (Optional)* tab opens. Set the *Enable jumpserver* toggle to 'On' if you want to provide this information. The following fields show:
    - **Host name or IP address:** The hostname or IP address of the jump server.
    - **Port Number:** The SSH port of the jump server.
    - **User ID:** The username for authentication on the jump server.
    - **Password:** The Password for authentication on the jump server.
10. Click **Done**.

Editing an SSH connection

After you create your connection, your can edit the connection. To edit a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **SSH** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the SSH connections page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.
5. Edit your connection. Click **Next** to go through the connection configurationn pages. Click **Save** when you are done editing.

Deleting an SSH connection

If you no longer need your SSH connection and want to delete it entirely, you can delete the connection from the console.

To delete a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **SSH** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the SSH connections page, click the options menu (three vertical dots) for the connection that you want to delete and click **Delete**.
5. Enter the name of the connection to confirm that you want to delete your connection. Then, click **Delete**.

Your connection is deleted.

Creating Zabbix connections

A Zabbix connection collects different types of metrics, including live and historical data, to monitor and analyze application performance. It gathers metrics at scheduled intervals from the Zabbix server, and sends data to the gRPC server as cloud events. IBM Cloud Pak for AIOps uses this metric data to identify deviations from the normal behavior and report anomalies. The AI algorithm that analyzes Zabbix data is Metric anomaly detection.

For more information about the Zabbix connections, see the following sections:

- Before you begin
- Creating Zabbix connections
- Enabling and disabling Zabbix connections
- Editing Zabbix connections

- Deleting Zabbix connections

For more information about HTTP headers for the various credential types, see HTTP headers for credential types.

Before you begin

Before you create a Zabbix connection, complete the following steps:

- Install Zabbix v6.0
- Add and configure hosts in the Zabbix server

For more information about installing Zabbix and setting up host, see the Installation and the Quickstart sections in the Zabbix documentation.

To create a Zabbix connection, you must have the following information:

- Endpoint
- API token

Creating Zabbix connections

To create a Zabbix connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Data and tool connections* page, click **Add connection**.
4. From the list of available connections, find and click the **Zabbix** tile.

   Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Connect**.
6. On the *Add connection* page, define the general connection details:

   - **Name**: Create a unique name for your connection.

     Note: Currently, the connection name field only supports alphanumeric characters and excludes spaces and other special characters.
   - **Description (optional)**: Provide a description for your connection.
   - **Endpoint**: Both http and https urls will be supported by the endpoint (for example, http://<Zabbix-server-hostname>/zabbix/api_jsonrpc.php or https://<Zabbix-server-hostname>/api_jsonrpc.php). In this case you can enter the following endpoint for your connection:

     `<protocol>://<Zabbix-server-hostname>/zabbix/api_jsonrpc.php`

     where **`protocol`** is HTTP or HTTPS and **`zabbix`** is the path to **`api_jsonrpc.php`** file, which is located in the front-end directory.
   - **Authentication type**: Select between **Username/Password** or **API token**.
     - **Username**: Enter a username for the Zabbix connection.
     - **Password**: Enter a password for the Zabbix connection.
     - **API token**: Enter an API token for the connection.
   - **Test Connection**: Click **Test connection**.

     Note: After you click **Test connection**, wait for the test to succeed before you continue to the next step. It might take some time. If the test connection fails, verify your credentials, and ensure that the server is up and running. However, you can still move to the metric selection page.

- **Deployment options**: Select one of the options, **Local** or **Remote**.
    - **Local**: This option deploys the connection in the same cluster where IBM Cloud Pak for AIOps is installed.
    - **Remote**: This option deploys it in your chosen location. After you add the connection, you get a bootstrap command to finish the deployment.

7. Click **Next**.

8. Enter *Collect metric data* information:

    - **Enable data collection**: Toggle the switch to the on position. However, if you disable data flow, a new connection is created, but metrics are not collected.

    - **Select templates**: You can either select all templates or choose the templates that you want to apply to the hosts. Only templates that have metrics will be shown in the UI.

        - To select all templates, click **Select all templates**.
        - To clear all selection, click **Unselect all templates**.
        - To choose the required templates, select the checkbox of the templates you require from the drop-down list.

      Note: You must select templates appropriate to the hosts. This is because if an item from the template is not available for the host, the connector cannot pull data for that item.

      For more information about performance considerations for metric data collection, see Performance considerations for metric data collection.

    - **Search and select the required metrics**: After the templates are loaded, the filter table displays all the associated built-in metrics of the selected templates. These metrics are preselected. You can search through and change the selection, if required.

        - To select a metric, click the checkbox of the required metric.
        - To remove a selected metric, clear its corresponding checkbox.
        - To select or clear all selected metrics, click **Built-in metrics name**. Alternatively, you can click **Select all** or **Unselect all** on the filter table header.
        - To change the size of the page that lists the built-metrics, select a number from the **Items per page** list at the end of the table.
        - To navigate through the pages to see all the metrics, click the next page or previous page arrows.
    - **Hosts**: The hosts are preselected on the load itself. To undo the selection, clear the checkbox of the hostname from the drop-down list.

      Note: If any of the hosts are not available, a notification is displayed. Unselect the hosts that are not available or ensure that all hosts are working in the Zabbix target system before you create your Zabbix connection. If a host goes down after you create the connection, edit and remove the host in your connection.

      Note: Zabbix organizes hosts based on their IP address. Therefore, if two hosts have the same IP address, then selecting one of these hosts results in both hosts getting selected.

    - **Base parallelism**: Specify the number of requests that can run in parallel. The default value is 3.

    - **History Span (minute)**: Enter a value in minutes. The value represents the time interval for live or historical data collection. The default value is 1.

    - *Select historical or live data collection*:

    Note: If you choose to collect only live data, you cannot collect historical data after the live data collection starts.

    - **Live data collection**: To collect live data, complete the following steps:

        - **Only collect live data once the connection is established**: Select the checkbox.
        - **Connection interval in minutes for live metrics (1-60)**: Enter a value in minutes. The value represents the frequency at which data is sampled for live metrics. The default value is 5.
    - **Historical data collection**: If you do not select live data collection, then historical data metrics are automatically collected. To collect historical data, complete the following steps:

        - **Start date and time**: Select the start date from the calendar and enter the time in the *hh:mm* format (hours and minutes).

Note: The start date must not exceed 31 days from the present as the maximum time period for historical data collection is 31 days. The recommended time period is two weeks.

- **Time zone**: Select your time zone from the drop-down list.
- **End date and time**: Click **Add end date** and select a date from the calendar. Also, enter the time in the *hh:mm* format (hours and minutes). If you want to remove an end date, click **Remove end date**.

Note: If you do not set an end date, IBM Cloud Pak for AIOps collects historical data from the specified start date and continues with live data collection. Therefore, you must specify the connection interval for live metrics.

- **Connection interval in minutes for live metrics (1-60)**: Enter a value in minutes. The value depicts the frequency at which data is sampled for live metrics. The default value is 5.

9. Click **Done**.

You created a Zabbix connection in your instance.

Enabling and disabling Zabbix connections

If you didn't enable your data collection during creation, you can enable your connection afterward. You can also disable a previously enabled connection the same way. To enable or disable a created connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. On the *Manage connections* tab of the *Data and tool connections* page, click the **Zabbix** connection type.
4. Click the connection that you want to enable or disable.
5. Set **Enable data collection** to **On** or **Off** to enable or disable log data collection. Disabling data collection for a connection does not delete the connection.

You enabled or disabled your connection. For more information about deleting a connection, see Deleting Zabbix connections.

Editing Zabbix connections

After you create your connection, your can edit the connection. For example, if you want to change the time interval of how often your job runs, you can edit it. To edit a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Zabbix** connection type on the *Manage connections* tab of the *Data and tool connections* page.
4. On the Zabbix connections page, click the name of the connection that you want to edit. Alternatively, you can click the options menu (three vertical dots) for the connection and click **Edit**. The connection configuration opens.
5. Edit your connection. Click **Next** to go through the connection configurationn pages. Click **Save** when you are done editing.

You edited your connection.

Deleting Zabbix connections

If you no longer need your zabbix connection and want to delete it entirely, you can delete the connection from the console.

To delete a connection, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
3. Click the **Zabbix** connection type on the *Manage connections* tab of the *Data and tool connections* page.

4. On the Zabbix connections page, click the options menu (three vertical dots) for the connection that you want to delete and click **Delete**.
5. Enter the name of the connection to confirm that you want to delete your connection. Then, click **Delete**.

Your connection is deleted.

Delay configuration in data connections

In live data mode the connector might need a delay to offset the query time windows to provide a time buffer and prevent retrieving partial real-time data.

The following table lists the environment variables for various integrations:

Integration environment variables

Integration	Environment variable for Delay
Custom	CUSTOM_QUERY_DELAY
ELK	ELK_QUERY_DELAY
Falcon LogScale	HUMIO_QUERY_DELAY
LOGDNA	LOGDNA_QUERY_DELAY
PagerDuty	PAGERDUTY_QUERY_DELAY
Splunk	SPLUNK_QUERY_DELAY

To add delay settings in the Flink, open the configuration by using the following commands:

- For PagerDuty:

  ```
  oc edit eventprocessor cp4waiops-eventprocessor
  ```

- For Falcon LogScale, Custom, LogDNA, ELK, and Splunk:

  ```
  oc edit deployment aimanager-aio-controller
  ```

  In the configuration, edit the env section.

  For example, set a 60-seconds delay for Splunk integration:

  ```
  env:
   - name: SPLUNK_QUERY_DELAY
     value: "60"
  ```

Note:

- The values that are set in the configuration must be in seconds.
- For existing connections to pick up delay values, you must save the connections again.
- Delays are not applicable to historical mode connections.

Mapping data from incoming sources

With any connection, you can specify your own mappings to the standard IBM Cloud Pak for AIOps values to optimize search performance for any data type.

For each type of connection, you can edit which fields map to the standard values. By default, each connection type (including custom connections) uses a default field mapping based on what the source data typically looks like. When you map source fields, you're replacing or supplementing those values with something else in your logs.

Mapping data types

- Mapping log data
- Mapping event data (Only available through a Kafka connection)
- Mapping similar ticket data from ServiceNow

Cleaning mapped data by using regular expressions

While you can use the *Field mapping* section of the connection page to customize field mappings to data values, you can also clean incoming data from your connection here. You can use regular expressions in the mapping field to strip formatting if you know that your data comes formatted in a specific way that affects training. For example, if your log files are always prefixed by the date and time, your training includes that prefix. As a result, IBM Cloud Pak for AIOps flags items that are also prefixed with the same timestamp. The following example illustrates how you can use the *Field mapping* section to clean your message prefixes.

Note: Using regular expressions in the *Field mapping* section is not limited to the following example. You can affect change on any mapped field in the same manner.

In this example, your LogDNA output produces the following timestamp at the beginning of every log entry:

```
Mar  9 03:30:01 TRACE [Controller id=2] Leader imbalance ratio for broker 2 is 0.0
(kafka.controller.KafkaController)
```

You can add `custom_regex` immediately following the mapping that you want to affect to exclude entries that match your proposed regular expression. In this example, you want to exclude timestamps that look like `Mar 9 03:30:01`. You can write a regular expression that follows the formatting of "MMM D HH:MM:SS" to ignore the month value to ensure that your training applies to all months. The following regular expression excludes the timestamp in the example log:

```
^[A-Z][a-z][a-z]\s+([1-9]|[1-2][0-9]|3[0-1])\s+[0-9][0-9]:[0-9][0-9]:[0-9][0-9]\s
```

Note: You can use AND (`&`) and OR operators (`|`) in combination with regular expressions to exclude more patterns. You can implement these operators in the *Field mapping* section by including the following content:

```
{
  "codec": "logdna",
  "message_field": "message",
  "custom_regex": [
    "^[A-Z][a-z][a-z]\\s+([1-9]|[1-2][0-9]|3[0-1])\\s+[0-9][0-9]:[0-9][0-9]:[0-9][0-9]\\s"
  ],
  "log_entity_types": "container",
  "instance_id_field": "_app",
  "timestamp_field": "_ts"
}
```

Important: If you previously trained your AI models with a different mapping, you must delete and retrain your models with the new mapping in place. For more information about altering models, see Configuring AI models.

Mapping log data from incoming sources

With any connection, you can specify your own mappings to the standard IBM Cloud Pak for AIOps values to optimize search performance for any data type.

For each type of connection (Custom, ELK, Falcon LogScale, Kafka, LogDNA, and Splunk), you can edit which fields map to the standard values. By default, each connection type (including custom connections) uses a default field mapping based on what the source data typically looks like. When you map source fields, you're replacing or supplementing those values with something else in your logs.

Mapping log anomaly data

The Log Anomaly service detects abnormal activities from system logs, which help Site Reliability Engineers (SREs) to better react to issues. Two types of anomalies are considered: *quantitative anomalies* (where constant linear or quantitative relationships are broken) and *sequential anomalies* (where a log sequence deviates from normal patterns of program flows).

IBM Cloud Pak for AIOps ingests logs from standard or custom data sources. Raw logs from supported standard data sources are transformed to a normalized format, as shown in the following example. Data from custom data sources must be transformed into normalized format first to be used in a connection (or eventually for AI model training).

Log anomaly raw data

To maintain the structure of the log anomaly data in the Elastic database, you must transform raw log anomaly data into a normalized format. The following examples illustrate raw JSON log anomaly data that comes from the ELK Stack, Falcon LogScale, and Splunk:

ELK Stack raw data example


```
{"_index":"syslog-console-example.com-
20201018","_type":"_doc","_id":"t1IaPXUBuJIWpBPwb1fG","_score":null,"_source":
{"MESSAGEID":"DSNJ001I","sysplexName":"MCMPLEX1","JOBNAME":"DC33MSTR","TEXT":"DSNJ001I  -DC11
DSNJW307 CURRENT COPY 2 ACTIVE LOG\nDATA SET IS
DSNAME=DSN11.LOGCOPY2.DS02,\nSTARTRBA=00000000000C20358000,ENDRBA=00000000000C2078FFFF\"\n","R
OUTECODE":"80000000000000000000000000000000","SYSPLEX_SYSTEM_JOB":"MCMPLEX1.ABCLP20.DC33MSTR",
"SYSPLEX_SYSTEM_CICS":"MCMPLEX1.ABCLP20","path":"SYSLOG","rcd":"MC","ASID":"0043","systemName"
:"ABCLP20","JOBNUM":"STC02552","TIMESTAMP":"20292 15.05.38.160
-0400","host":"ABCLP20.svl.ibm.com","@version":"1","FLAGS":"00","seq":
{"c":"14469","w":"4"},"DESCRIPTOR":"00000090","timeZone":"-0400","message":"MC,0043,20292
15.05.38.160 -0400,ABCLP20 ,STC02552,
,80000000000000000000000000000000,00000090,DC33MSTR,00,\" DSNJ001I  -DC11 DSNJW307 CURRENT
COPY 2 ACTIVE LOG\n DATA SET IS DSNAME=DSN11.LOGCOPY2.DS02,\n
STARTRBA=00000000000C20358000,ENDRBA=00000000000C2078FFFF\"\n","inputsequence":"20201018190543
703:000000","tags":"pipeline","SMFID":"ABCLP20 ","@timestamp":"2020-10-18T15:05:38.160-
04:00","port":6911,"CONSOLE":"        ","sourceType":"syslog-
console","SYSPLEX_SYSTEM_DB2":"MCMPLEX1.ABCLP20","sourceName":"ABCLP20-
SYSLOG","SYSPLEX_SYSTEM":"MCMPLEX1.ABCLP20"},"sort":[1603047938160]}
```


Falcon LogScale (previously named Humio) raw data example


```
{"kubernetes.namespace_name":"myhumio","@timestamp":1603445464206,"kubernetes.host":"worker.ho
st.example.com","kubernetes.container_hash":"docker.io/value/enterprise-
kafka@sha256:bb3d4572b51490b7e9fc2431125ae963bf7cdc4cd30ff4575b1e529e3e53a1e6","kubernetes.doc
ker_id":"39e5cba870732269940daa8378ee2eab52adffa93dd6c5ccf15246e70b115584","kubernetes.labels.
app":"cp-kafka","kubernetes.container_name":"cp-kafka-
broker","kubernetes.container_image":"docker.io/value/enterprise-
kafka:5.2.1","kubernetes.labels.controller-revision-hash":"humio-example-kafka-
66c4fb4d64","kubernetes.annotations.k8s_v1_cni_cncf_io/networks-status":"[{\n    \"name\":
\"openshift-sdn\",\n    \"interface\": \"eth0\",\n    \"ips\": [\n        \"10.254.20.30\"\n
],\n    \"dns\": {},\n    \"default-route\": [\n        \"10.254.20.1\"\n
]\n}]","kubernetes.annotations.prometheus_io/scrape":"true","@timezone":"Z","@rawstring":"2020
-10-23T09:31:04.206512759+00:00 stdout F [2020-10-23 09:31:04,206] TRACE [Controller id=2]
Leader imbalance ratio for broker 2 is 0.0
(kafka.controller.KafkaController)","kubernetes.labels.statefulset_kubernetes_io/pod-
name":"humio-example-kafka-
2","@id":"yI1o4C0cV5i8oGU4TkcbavzO_587_147_1603445464","#type":"unparsed","kubernetes.annotati
ons.openshift_io/scc":"restricted","kubernetes.labels.release":"humio","#repo":"kubernetes","k
ubernetes.pod_name":"humio-example-kafka-2","kubernetes.pod_id":"7a4b0b92-1254-4335-ae29-
b1f548c8e21d","#humioAutoShard":"1","kubernetes.annotations.prometheus_io/port":"5556"}
```


Splunk raw data example


```
{"_bkt":"_internal~53~283A637C-D363-4549-A819-
E8175BC73650","_cd":"53:2848711","_indextime":"1603220666","_raw":"127.0.0.1 - splunk-system-
user [20/Oct/2020:12:04:26.272 -0700] \"GET /services/saved/searches?count=0 HTTP/1.1\" 200
1861412 - - - 162ms","_serial":"31263","_si":
["abc.example.com","_internal"],"_sourcetype":"splunkd_access","_subsecond":".272","_time":"20
20-10-20T12:04:26.272-
07:00","host":"abc.example.com","index":"_internal","linecount":"1","source":"/root/mj/mjsplun
k/splunk/var/log/splunk/splunkd_access.log","sourcetype":"splunkd_access","splunk_server":"abc
.example.com"}
```


Normalized log data example

The following sample illustrates the result of mapping data from an external log source for use with IBM Cloud Pak for AIOps:

```
{
    "timestamp": 1655991641000,
    "instance_id": "calico-node-jn2d2",
    "application_group_id": "1000",
    "application_id": "1000",
    "features": [],
    "meta_features": [],
    "message": "[64] ipsets.go 254: Resyncing ipsets with dataplane. family=\"inet\"",
    "entities": {
        "pod": "calico-node-jn2d2",
        "cluster": null,
        "container": "calico-node",
        "node": "kube-bmgcm5td0stjujtqv8m0-ai4itsimula-wp16cpu-00002c34"
    },
    "type": "StandardLog"
}
```

Log anomaly normalized attributes

Required attributes expected as input for the pipeline

Attribute	Description
`timestamp`	Epoch Timestamp of the event in the log entry. The epoch time should be in milliseconds (13 digit) and not in seconds (10 digit).
`utc_timestamp`	The `utc_timestamp` of the event in the log entry. The format should be in `yyyy-MM-dd HH:mm:ss.SSS` format. The value of this field should be equivalent for the epoch value of the `timestamp` field.
`instance_id`	Logs can be trained based on certain attributes, like the application to which the log entry belongs, the hosts from which the log is generated, or the pods from which the log is generated. `instance_id` is the value of the attribute that is chosen to be the key upon which the models are built. Attributes chosen to be `instance_id` are specified in the ingest configuration during onboarding.
`application_id`	Application from which the log entry was generated. It can be the same as `instance_id` if the `instance_id` is chosen to be the application
`features`	Contains log templates
`message`	Log message for the log entry. It can be an error message, a normal information message, or a debug message. A good combination of message types yields the best results. For example, if the logs are only error messages or if no error messages appear in the logs, the quality of the anomaly detection suffers.
`entities`	Entities are provided for better event attribution. They provide clues to connect information from multiple data sources such as events, logs, and incidents, and also are used to globalize faulty components. Entities can be infrastructure-related attributes like node name, hostname, IP address, pod name, container name, or cluster name. These attributes are specified as a comma-separated list in the configuration during onboarding. The `entities` attribute must, at least, contain an empty field: `"entities": {}`.

Static attributes

Attribute	Description
`application`	Application that the IBM Cloud Pak for AIOps instance is monitoring. Provided as part of the initial configuration during onboarding
`application_group_id`	Supports multi-tenancy. Provided as part of the initial configuration during onboarding
`type`	Hardcoded to the log

Attributes analyzed and populated in the pipeline

Attribute	Description
`meta_features`	Contains windowing details

Examples

Review the following examples to view how to map log data.

Example: Falcon Logscale field mapping

- **`message_field`** maps to **`@rawstring`**
- **`log_entity_types`** maps to **`kubernetes.container_name`**
- **`instance_id_field`** maps to **`kubernetes.container_name`**
- **`timestamp_field`** maps to **`@timestamp`**

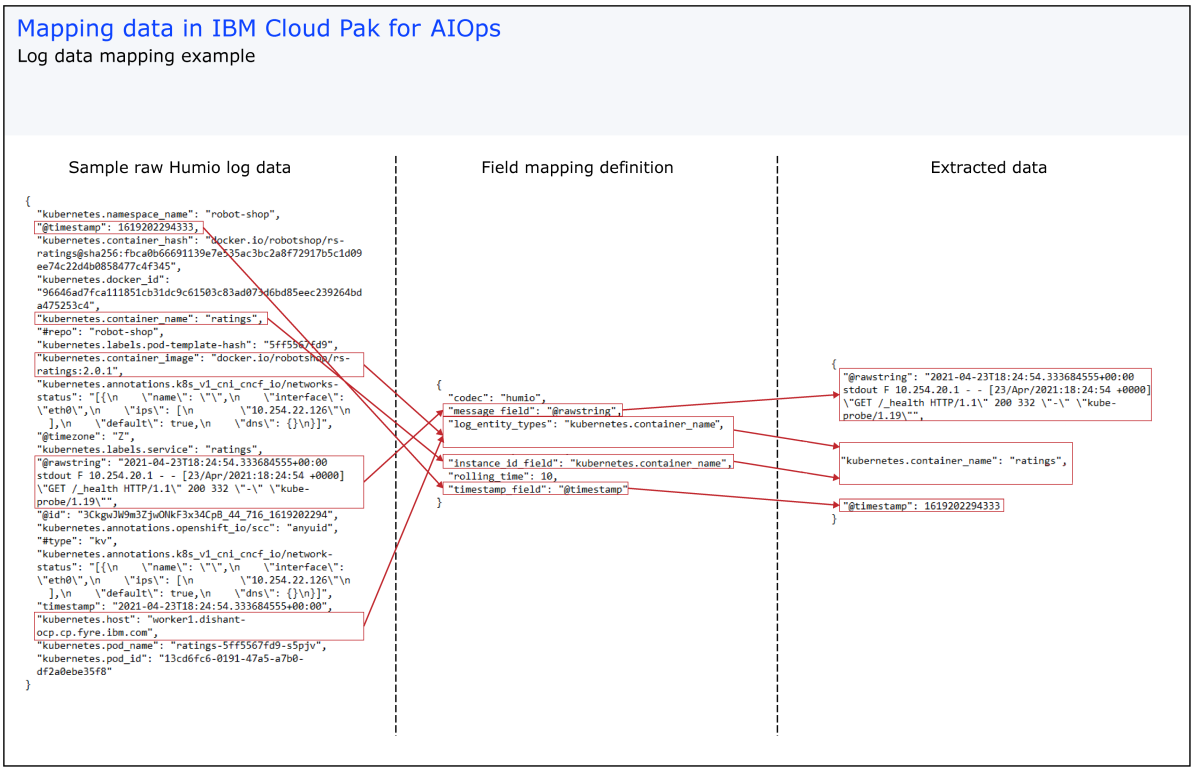


Figure. Falcon Logscale mapping example

All of the values in this Falcon LogScale field mapping correspond to entries in the Falcon LogScale raw data. In this example, the following values map to one another in the Falcon LogScale raw log.

Important: You must specify the corresponding codec in your field mapping when you create your connection. If you do not, the mapping fails. In the preceding example, the mapping specifies **`humio`** as the codec value.

You can define mappings when you create your connection. The basic building blocks of mappings are similar for most connection types. For logs, the **`codec`**, **`rolling_time`**, **`instance_id_field`**, **`log_entity_types`**, **`message_field`**, and **`resource_id`** mappings typically correspond to values that you might find in your log data. The following table describes typical mapping fields for logs:

Typical mapping values for log connections

Field Name	Description
`codec`	The type of connection for your incoming data. You must specify the codec value for non-log connections too. For example, you must specify **`humio`** for a Falcon Logscale connection mapping, or **`servicenow`** for a ServiceNow connection.
`rolling_time`	The recurring time interval to capture data from in seconds. The default value is 10 seconds.
`instance_id_field`	The field name in the incoming data that contains the application name. This value is used to identify where your log data is coming from. If you choose an overly specific value, it might lead to flooding the pipeline with irrelevant log data. Ideal candidates for mapping this field include: **`kubernetes.namespace_name`**, **`kubernetes.host`**, or **`kubernetes.container_name`**.
`log_entity_types`	The field name in the incoming data that contains the entity to extract and analyze. The value provides information about the app or service where the log data comes from. Ideal candidates for mapping this field include: **`kubernetes.namespace_name`**, **`kubernetes.host`**, or **`kubernetes.container_name`**.
`message_field`	The field name in the incoming data that contains the log message. For example, **`@rawstring`**.

Field Name	Description
`resource_id`	The field name in the incoming data, which contains the specific component that produces the log message. For example, if you deploy your application through a Kubernetes deployment with four replicas, then the `resource_id` field name contains the specific pod name that produces the log message. This information helps to narrow down the component that must be analysed on detection of an anomaly in your application.

Important: Be mindful when you select mapping values for your `instance_id_field` and `log_entity_types` fields. `instance_id_field` identifies the app or service where your messages come from. `instance_id_field` acts almost like the application name in that respect. `log_entity_types` provides information about the app or service, like the name of the pod, the image used in the pod, the node of the pod, and more. If you chose values that are too specific, you might gather too much information from irrelevant pods or projects (namespaces).

Example: Log anomaly output schema

```
{
    "application_group_id": "1",
    "application_id": "1",
    "timestamp":1655991641000,
    "utc_timestamp":"2022-06-23 13:40:41.000"
    "type": "log-anomaly-detection",
    "alert": {
      "alert_id": "6751a428-5827-11ea-9f3a-c4b301ca8457",
      "title": "NETWORK <*> end connection <*> (<*> <*> now open) NETWORK [listener]
connection accepted from <*> #<*> (<*> connections now open)",
      "text": "NETWORK <*> end connection <*> (<*> <*> now open) NETWORK [listener] connection
accepted from <*> #<*> (<*> connections now open)",
      "created_at_utc_timestamp": "2020-02-25T23:34:44.874301",
      "severity": 3,
      "source": {
        "source_name": "anomaly-detector",
        "source_url": "",
        "source_alert_id": "6751a428-5827-11ea-9f3a-c4b301ca8457",
        "source_team": {
          "id": "",
          "name": "",
          "web_url": ""
        },
        "source_application": {
          "id": "ts-assurance-mongo",
          "name": "ts-assurance-mongo",
          "web_url": ""
        }
      },
      "features": [
        {
          "id": "log_anomaly_data",
          "value": {
            "id": "6751a428-5827-11ea-9f3a-c4b301ca8457",
            "start_timestamp": 1578669070000,
            "end_timestamp": 1578669080000,
            "log_anomaly_data": {
              "detected_at": 1582702484874.0,
              "source_application_id": "ts-assurance-mongo",
              "log_anomaly_confidence": 1.0,
              "template_sequence": [
                "NETWORK [listener] connection accepted from <*> #<*> (<*> connections now
open)",
                "NETWORK [listener] connection accepted from <*> #<*> (<*> connections now
open)",
                "NETWORK <*> end connection <*> (<*> <*> now open)",
                "NETWORK <*> end connection <*> (<*> <*> now open)"
              ],
              "template_list": [
                "[ENTSV:ERRR] <*> <*> <*> <*> <*>",
                "Open Entities Serve Exception received. Max retries: <*>, Current retry <*>",
                "Exception during getNLQuestionClasses() call",
                "Exception invoking mmesh.EntityPredictionService/predictTokenizedDocument
method for model ids <*>",
                "Error invoking mmesh.EntityPredictionService/predictTokenizedDocument method
```

```
on model <*> INTERNAL: <ENT02360739W> Exception caught: Authentication failed.",
                "applyModel",
                "Error on query to NLU engine.",
                "Error occurred while processing a message v2 request"
              ],
              "count_vector": [
                2,
                2,
                0,
                0
              ]
            },
            "entity": {
              "pod": [
                {
                  "name": "ts-assurance-mongo-54b557bfbc-7txwh",
                  "counts": 4
                }
              ],
              "cluster": [
                {
                  "name": "null",
                  "counts": 4
                }
              ],
              "container": [
                {
                  "name": "ts-assurance-mongo",
                  "counts": 4
                }
              ],
              "node": [
                {
                  "name": "kube-bmgcm5td0stjujtqv8m0-ai4itsimula-wp16cpu-00002d54",
                  "counts": 4
                }
              ]
            }
          }
        }
      ],
      "resolved_at_utc_timestamp": null
    },
    "meta_features": [
      {}
    ]
  }
```

Example: Normalized output schema

The preceding example adheres to the normalized output schema for IBM Cloud Pak for AIOps, which can resemble the following example output schema.

```
{
    "$id": "incidents-schema.json",
    "$schema": "http://json-schema.org/draft-06/schema",
    "title": "incidents",
    "type": "object",
    "properties": {
        "application_group_id": {
            "description": "Unique identifier for the tenant which owns the application being
monitored (defined by IBM Cloud Pak for AIOps)",
            "type": "string"
        },
        "application_id": {
            "description": "Unique identifier of the application being monitored (defined by
IBM Cloud Pak for AIOps)",
            "type": "string"
        },
        "incident_id": {
            "description": "The unique id for this incident created by IBM Cloud Pak for
AIOps",
```

```
            "type": "string"
        },
        "title": {
            "description": "The short title for the given incident (SN short_description)",
            "type": "string"
        },
        "description": {
            "description": "The description for the given incident (SN description)",
            "type": "string"
        },
        "created_at": {
            "description": "The timestamp the incident was created at (SN sys_created_on).",
            "type": "string"
        },
        "updated_at": {
            "description": "The timestamp the incident was last updated (SN sys_updated_on).",
            "type": "string"
        },
        "resolved_at": {
            "description": "The timestamp the incident was resolved (SN resolved_at).",
            "type": "string"
        },
        "closed_at": {
            "description": "The timestamp the incident was closed (SN closed_at).",
            "type": "string"
        },
        "started_at": {
            "description": "The timestamp the disruption (which caused this incident to be
created) was started at (SN opened_at).",
            "type": "string"
        },
        "business_duration_ms": {
            "description": "The duration in ms the incident. (SN business_duration)",
            "type": "integer"
        },
        "severity": {
            "description": "The severity, as set by the client, of the incident on a scale
from 1 (highest) to 5 (lowest) (SN severity)",
            "type": "integer",
            "enum": [1,2,3,4,5]
        },
        "priority": {
            "description": "The priority, as set by the helpdesk, of the incident on a scale
from 1 (highest) to 5 (lowest) (SN priority)",
            "type": "integer",
            "enum": [1,2,3,4,5]
        },
        "impact": {
            "description": "The impact of the incident on a scale from 1 (highest) to 5
(lowest) (SN impact)",
            "type": "integer",
            "enum": [1,2,3,4,5]
        },
        "state": {
            "description": "The current state of the incident (SN state)",
            "type": "string",
            "enum": ["new", "in_progress", "resolved", "closed"]
        },
        "source": {
            "type": "object",
            "description": "The source that the incident was read from",
            "properties": {
                "source_name": {
                    "description": "The display name for the incident system this incident was
read from (such as ServiceNow).",
                    "type": "string"
                },
                "source_url": {
                    "description": "The url for the incident system this incident was read
from (such as https://....).",
                    "type": "string"
                },
```

```
                "source_incident_id": {
                    "description": "The id for this incident in the original source (SN
number)",
                    "type": "string"
                },
                "source_application_id": {
                    "description": "Id of the application that had the disruption (SN
cmdb_ci)",
                    "type": "string"
                }
            },
            "required": [
                "source_name",
                "source_incident_id"
            ]
        },
        "comments": {
            "description": "The set of comments and work notes for the incident",
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "comment_text": {
                        "description": "The text for the given comment (SN
comments_and_work_notes).",
                        "type": "string"
                    }
                },
                "additionalProperties": false,
                "required": [
                    "comment_text"
                ]
            },
            "minItems": 0
        },
        "related_incidents": {
            "description": "The set of incidents that are related (SN parent_incident &
child_incidents)",
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                    "source_incident_id": {
                        "description": "The id for the related incident in the source
system.",
                        "type": "string"
                    },
                    "relationship": {
                        "description": "The relationship to this incident, namely parent child
or similar",
                        "type": "string",
                        "enum": ["parent","child","similar"]
                    }
                },
                "additionalProperties": false,
                "required": [
                    "comment_text"
                ]
            },
            "minItems": 0
        },
        "resolution": {
            "type": "object",
            "description": "The source that the incident was read from",
            "properties": {
                "rca_id": {
                    "description": "The id of the rca (problem) ticket that was created in
response to this incident (SN problem_id).",
                    "type": "string"
                },
                "resolution_summary": {
                    "description": "The summary of what was done to resolve the issue (SN
```

```
close_notes).",
                    "type": "string"
                }
            },
            "required": [
                "resolution_summary"
            ]
        },
        "features": {
            "description": "The extended set of features related to the incident",
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                },
                "additionalProperties": true,
                "required": [
                ]
            },
            "minItems": 0
        },
        "required": [
            "application_group_id",
            "application_id",
            "incident_id",
            "created_at_utc_timestamp",
            "title",
            "description",
            "severity",
            "state",
            "source"
        ],
        "meta_features": {
            "description": "The extended set of features related to the message",
            "type": "array",
            "items": {
                "type": "object",
                "properties": {
                },
                "additionalProperties": true,
                "required": [ ]
            },
            "minItems": 0
        }
    }
}
```

Mapping event data for a Kafka connection

With a Kafka connection, you can gather data for both logs and events. If you are gathering event data, you can specify your own mappings to the standard IBM Cloud Pak for AIOps values.

By default, the Kafka connection uses a default field mapping based on what the source data typically looks like. When you map source fields, you're replacing or supplementing those values with something else in your logs.

Important: Values mapped to the `alert.type.classification` field in the Cloud Pak for AIOps data model are especially significant. Inappropriate values assigned to this field at the time of mapping has a negative impact on several downstream activities, such as Golden Signal classification, incident creation, and recommended policies. Classification should contain information summarizing the event type, for example, CPU Utilization, Node Down, or the actual metric name itself.

Mapping event grouping data

Event grouping receives streams of events from a Kafka connection, and continuously groups events in real time. This grouping process uses various techniques such as deduplication, temporal clustering, and more. The event grouping service interacts with other services to further provide localization, average severity, pattern recognition, diagnosis, and explanation.

The event grouping service takes normalized data through the connection from an event data source, such as PagerDuty. That data is then used to fit into a single structure that the event grouping service uses to derive new information, such as a topology. Event grouping aims to suppress the number of events to help you diagnose problems with relevant events. This grouping increases the efficiency of handling incidents, and associates the event with other information (such as logs or metrics).

Event raw data

To maintain the structure of the event data in the Elasticsearch database, you must map and transform raw event data into a normalized format. This normalized data can then be used in AI training.

The following raw JSON data shows example event data from a PagerDuty data source:


```
{"id": "PBDVRS5", "type": "alert", "summary": "[Kubernetes] Pods Available less than desired",
"self": "https://api.pagerduty.com/alerts/PBDVRS5", "html_url":
"https://example.pagerduty.com/alerts/PBDVRS5", "created_at": "2020-01-17T19:21:24-05:00",
"status": "resolved", "resolved_at": "2020-01-17T19:34:22-05:00", "alert_key": "3014911",
"suppressed": false, "service": {"id": "PTOQ55S", "type": "service_reference", "summary":
"simulation", "self": "https://api.pagerduty.com/services/PTOQ55S", "html_url":
"https://example.pagerduty.com/services/PTOQ55S"}, "severity": "critical", "incident": {"id":
"P6ZLD21", "type": "incident_reference", "summary": "[#29918643] [Kubernetes] Pods Available
less than desired", "self": "https://api.pagerduty.com/incidents/P6ZLD21", "html_url":
"https://example.pagerduty.com/incidents/P6ZLD21"}, "first_trigger_log_entry": {"id":
"RNSI4KGVTUAA2ZJ8IILWPF52PL", "type": "trigger_log_entry_reference", "summary": "Triggered
through the API", "self": "https://api.pagerduty.com/log_entries/RNSI4KGVTUAA2ZJ8IILWPF52PL",
"html_url":
"https://example.pagerduty.com/alerts/PBDVRS5/log_entries/RNSI4KGVTUAA2ZJ8IILWPF52PL"},
"body": {"contexts": [{"href": "https://cloud-
server.monitoring.cloud.example.com:443/#/alerts/343016", "text": "Link to Alert definition
that triggered the event", "type": "link"}, {"href": "https://cloud-
server.monitoring.cloud.example.com:443/#/events/notifications/1:604800/3014911/details",
"text": "Troubleshoot Event", "type": "link"}], "details": {"Value": "0.95; 1.0", "UTC":
"2020-01-18T00:19Z", "Timestamp": 1579306740000, "Subject": "[Kubernetes] Pods Available less
than desired is Triggered on kubernetes.namespace.name = default and
kubernetes.deployment.name = ts-admin-travel-service", "Source": "Sysdig", "Severity": "Low",
"Segment": "kubernetes.deployment.name = 'ts-admin-travel-service' and
kubernetes.namespace.name = 'default'", "Scope": "kubernetes.deployment.name = 'ts-admin-
travel-service' and kubernetes.namespace.name = 'default'", "Event ID": 3014911, "Condition":
"avg(timeAvg(kubernetes.deployment.replicas.available)) <
avg(timeAvg(kubernetes.deployment.replicas.desired))", "Body": "\n\n\nEvent
Generated:\n\nSeverity:         Low\n    Metric:\n    kubernetes.deployment.replicas.desired =
1\nSegment:\n    kubernetes.deployment.name = 'ts-admin-travel-service' and
kubernetes.namespace.name = 'default'\nScope:\n    Everywhere\n\nTime:             01/18/2020
12:19 AM UTC\nState:            Triggered\nNotification URL: https://cloud-
server.monitoring.cloud.example.com:443/api/oauth/openid/example/6f42245bn4ab5gc294c55b5785e74
ab9/ba4b7qqa-1fe9-47d1-9355-1i97d23a1855?
redirectRoute&#x3D;/events/notifications/1:2419200/3014911/details\n\n------\n\nTriggered by
Alert:\n\nName:         [Kubernetes] Pods Available less than desired\nDescription:  Not
enough pods running for deployment\nTeam:         Monitor Operations\nScope:\n
Everywhere\nSegment by:   kubernetes.namespace.name, kubernetes.deployment.name\nWhen:
avg(timeAvg(kubernetes.deployment.replicas.available)) <
avg(timeAvg(kubernetes.deployment.replicas.desired))\nFor at least: 10 min\nAlert URL:
https://cloud-
server.monitoring.cloud.example.com:443/api/oauth/openid/example/6f58425be4114ab295c55b3238e74
ab9/ba5n7gga-1fe9-47d1-9355-1p97d63a1855?redirectRoute&#x3D;/alerts/343016\n\n\n", "Alert
name": "[Kubernetes] Pods Available less than desired", "Alert description": "Not enough pods
running for deployment", "Alert ID": 343016}, "cef_details": {"client": "Sysdig",
"client_url": "https://cloud-
server.monitoring.cloud.example.com:443/#/events/notifications/1:604800/3014911/details",
"contexts": [{"type": "link", "text": "Link to Alert definition that triggered the event",
"href": "https://cloud-server.monitoring.cloud.example.com:443/#/alerts/343016"}, {"type":
"link", "text": "Troubleshoot Event", "href": "https://cloud-
server.monitoring.cloud.example.com:443/#/events/notifications/1:604800/3014911/details"}],
"dedup_key": "3014911", "description": "[Kubernetes] Pods Available less than desired",
"details": {"Value": "0.95; 1.0", "UTC": "2020-01-18T00:19Z", "Timestamp": 1579306740000,
"Subject": "[Kubernetes] Pods Available less than desired is Triggered on
kubernetes.namespace.name = default and kubernetes.deployment.name = ts-admin-travel-service",
"Source": "Sysdig", "Severity": "Low", "Segment": "kubernetes.deployment.name = 'ts-admin-
travel-service' and kubernetes.namespace.name = 'default'", "Scope":
"kubernetes.deployment.name = 'ts-admin-travel-service' and kubernetes.namespace.name =
```

```
'default'", "Event ID": 3014911, "Condition":
"avg(timeAvg(kubernetes.deployment.replicas.available)) <
avg(timeAvg(kubernetes.deployment.replicas.desired))", "Body": "\n\n\nEvent
Generated:\n\nSeverity:         Low\n    Metric:\n    kubernetes.deployment.replicas.desired =
1\nSegment:\n    kubernetes.deployment.name = 'ts-admin-travel-service' and
kubernetes.namespace.name = 'default'\nScope:\n    Everywhere\n\nTime:             01/18/2020
12:19 AM UTC\nState:            Triggered\nNotification URL: https://cloud-
server.monitoring.cloud.example.com:443/api/oauth/openid/example/5gabc35be4ab4cc294c55b5785e74
ab9/ba4b7nna-1fe9-47d1-9355-fm65d63a1855?
redirectRoute&#x3D;/events/notifications/1:2419200/3014911/details\n\n------\n\nTriggered by
Alert:\n\nName:         [Kubernetes] Pods Available less than desired\nDescription:  Not
enough pods running for deployment\nTeam:         Monitor Operations\nScope:\n
Everywhere\nSegment by:   kubernetes.namespace.name, kubernetes.deployment.name\nWhen:
avg(timeAvg(kubernetes.deployment.replicas.available)) <
avg(timeAvg(kubernetes.deployment.replicas.desired))\nFor at least: 10 min\nAlert URL:
https://cloud-
server.monitoring.cloud.example.com:443/api/oauth/openid/example/6f86155be4ab4cc294c55b5785e74
ab9/ba4b7bba-1fe9-47d1-9355-ba97d63a1855?redirectRoute&#x3D;/alerts/343016\n\n\n", "Alert
name": "[Kubernetes] Pods Available less than desired", "Alert description": "Not enough pods
running for deployment", "Alert ID": 343016}, "message": "[Kubernetes] Pods Available less
than desired", "mutations": [], "version": "1.0"}, "type": "alert_body"}, "integration":
{"id": "PKKYNGU", "type": "generic_events_api_inbound_integration_reference", "summary":
"Sysdig-ai4it", "self": "https://api.pagerduty.com/services/PTOQ55S/integrations/PKKYNGU",
"html_url": "https://example.pagerduty.com/services/PTOQ55S/integrations/PKKYNGU"},
"privilege": null, "team": {"id": "PWY2HS1", "type": "team", "summary": "Project Zeno E2E",
"self": "https://api.pagerduty.com/teams/PWY2HS1", "html_url":
"https://example.pagerduty.com/teams/PWY2HS1"}}
```


Description of PagerDuty normalized attributes

The following table provides details about the attributes that are expected in the normalized data that results from mapping and transforming the raw PagerDuty data.

Normalized attributes for PagerDuty event data

PagerDuty attribute	Normalized attribute	Description
`created_at`	`timestamp`	Epoch timestamp of the event in the log entry
`created_at`	`utc_timestamp`	Timestamp of the event in the log entry, in the format `"yyyy-mm-ddTHH:MM:SSZ"`
`summary`	`alert.title`	Title of the entry created
`summary`	`alert.text`	Describes the condition and the affected managed object instance
`created_at`	`alert.created_at_utc_timestamp`	Time the entry was created
`resolved_at`	`alert.resolved_at_utc_timestamp`	Time the entry was resolved
`severity`	`alert.severity`	Indicates the severity level from 1 (indeterminate) to 5 (critical)
`html_url`	`alert.source.source_url`	URL of the event in PagerDuty
`id`	`alert.source.source_alert_id`	Internal PagerDuty ID assigned to the event
`team.id`	`alert.source.source_team.id`	Internal PagerDuty ID associated with the team that is assigned to the event. Team data is added to the original PagerDuty payload with an extra API call
`team.summary`	`alert.source.source_team.name`	Name of the team that is assigned to the event
`team.html_url`	`alert.source.source_team.web_url`	URL of the team in PagerDuty
`service.id`	`alert.source.source_application.id`	Internal PagerDuty ID associated with the service that is assigned to the event
`service.summary`	`alert.source.source_application.name`	Name of the service that is assigned to the event
`service.html_url`	`alert.source.source_application.web_url`	URL of the service in PagerDuty
`instance.id`	`alert.features.<id:pagerduty_incident>.value.id`	Internal PagerDuty ID associated with the incident that is assigned to the event `features` is an array. `incident.id` goes into the `value.id` field of the object with `id:pagerduty_incident`

PagerDuty attribute	Normalized attribute	Description
`instance.summary`	`alert.features.<id:pagerduty_incident>.value.name`	Name of the service that is assigned to the alert
`instance.html_url`	`alert.features.<id:pagerduty_incident>.value.web_url`	URL to the incident in PagerDuty
`<entire raw PagerDuty JSON>`	`alert.features.<id:root>.value`	The entire raw PagerDuty JSON goes into the value field of the object with `id:root`

Example of normalized PagerDuty event

```
{
  "application_group_id": "123abc",
  "application_id": "123abc",
  "timestamp": 1581531561.4586372,
  "utc_timestamp": "2020-02-12 18:19:21.458637",
  "type": "alert",
  "alert": {
    "alert_id": "7f83fb17-5a02-8c86-020d23cfe01c",
    "title": "A failure occurred invoking the provision callback on conversation",
    "text": "A failure occurred invoking the provision callback on conversation",
    "created_at_utc_timestamp": "2019-12-09T11:20:46-05:00",
    "resolved_at_utc_timestamp": null,
    "severity": 4,
    "source": {
      "source_name": "PagerDuty",
      "source_url": "https://<pagerdutyserver.com>/alerts/PBCAOOO",
      "source_alert_id": "PBCAOOO",
      "source_team": {
        "id": "P8FWLZZ",
        "name": "SRE - Assistant",
        "web_url": "https://<pagerdutyserver.com>/teams/P8FWLZZ"
      },
      "source_application": {
        "id": "P67C6KK",
        "name": "FirstTest",
        "web_url": "https://<pagerdutyserver.com>/services/P67C6KK"
      }
    },
    "features": [
      {
        "id": "pagerduty_incident",
        "value": {
          "id": "PGAC9NN",
          "name": "[#291477] A failure occurred invoking the provision callback on
conversation",
          "web_url": "https://<pagerdutyserver.com>/incidents/PGAC9NN"
        }
      },
      {
        "id": "root",
        "value": {
          "id": "PBCAOOO",
          "type": "alert",
          "summary": "A failure occurred invoking the provision callback on conversation",
          "self": "https://<pagerdutyapi>/alerts/PBCAOOO",
          "html_url": "https://<pagerdutyserver.com>/alerts/PBCAOOO",
          "created_at": "2019-12-09T11:20:46-05:00",
          "status": "triggered",
          "resolved_at": null,
          "alert_key": "mni-dedup-key-2-2019-12-09-17:20:34",
          "suppressed": false,
          "service": {
            "id": "P67C6KK",
            "type": "service_reference",
            "summary": "FirstTest",
            "self": "https://<pagerdutyapi>/services/P67C6KK",
            "html_url": "https://<pagerdutyserver.com>/services/P67C6KK"
```

```
          },
          "severity": "info",
          "incident": {
            "id": "PGAC9NN",
            "type": "incident_reference",
            "summary": "[#29147766] A failure occurred invoking the provision callback on
conversation",
            "self": "https://<pagerdutyapi>/incidents/PGAC9NN",
            "html_url": "https://<pagerdutyserver.com>/incidents/PGAC9NN"
          },
          "first_trigger_log_entry": {
            "id": "R4Y7P9Y8HTZ9VRPJ",
            "type": "trigger_log_entry_reference",
            "summary": "Triggered through the API",
            "self": "https://<pagerdutyapi>/log_entries/R4Y7P9Y8HTZ9VRPJ",
            "html_url":
"https://<pagerdutyserver.com>/alerts/PBCAOOO/log_entries/R4Y7P9Y8HTZ9VRPJ"
          },
          "body": {
            "contexts": [],
            "details": null,
            "cef_details": {
              "client": "curl",
              "contexts": [],
              "dedup_key": "mni-dedup-key-2-2019-12-09-17:20:34",
              "description": "A failure occurred invoking the provision callback on
conversation",
              "event_class": "Test",
              "message": "A failure occurred invoking the provision callback on conversation",
              "mutations": [],
              "service_group": "app-testing",
              "severity": "info",
              "source_component": "manual-test",
              "source_origin": "000.00.000.000",
              "version": "1.0"
            },
            "type": "alert_body"
          },
          "integration": {
            "id": "P02BNYY",
            "type": "events_api_v2_inbound_integration_reference",
            "summary": "AppCluster",
            "self": "https://<pagerdutyapi>/services/P67C6GI/integrations/P02BNYY",
            "html_url": "https://<pagerdutyserver.com>/services/P67C6GI/integrations/P02BNYY"
          },
          "privilege": null,
          "team": {
            "id": "P8FWLZZ",
            "type": "team",
            "summary": "SRE - Assistant",
            "self": "https://<pagerdutyapi>/teams/P8FWLZZ",
            "html_url": "https://<pagerdutyserver.com>/teams/P8FWLZZ"
          }
        }
      }
    ]
  },
  "meta_features": []
}
```


Performance considerations for metric data collection

When configuring a connection in IBM Cloud Pak for AIOps, which collects metric data such as AppDynamics, AWS CloudWatch, Dynatrace, Instana, New Relic, Splunk, and Zabbix connectors, it is important to consider the amount of data on the target system when selecting metrics.

If you have an exceptionally large amount of data on the selected target system, there can be relatively higher resource usage from collecting data from that system when compared to a system with less data.

Connector filtering capabilities table

The types of filtering available for different connectors are indicated here. No target system supports all filtering options, which is why they are different for each connector.

Connector filtering capabilities matrix

Metric Connector Name	Metrics Filtering	Host Filtering	Namespace Filtering	Application Filtering	Technology Filtering	Index Filtering
AppDynamics	X			X		
AWS CloudWatch	X		X			
Dynatrace	X				X	
Instana	X				X	
New Relic	X			X		
Splunk	X	X				X
Zabbix	X	X			X	

For target systems with a high amount of data, consider the following actions to ensure your connection works as expected:

- Do not select all metrics. Instead, review the documentation for the target system you are pulling data for and select specific metrics that are useful. Not all metrics are suited to all use cases.
- Do not select all hosts for a particular connection. If you have thousands of hosts, it can be better to create multiple connections instead of one connection with all your hosts.
- Do not create more than 10 connections on a small cluster or 20 connections on a large cluster at any one time. The more connections that you have running simultaneously, the more computing resources are needed.
- If the Kafka persistent volume claims (PVCs) near capacity, increase each of the PVCs by an extra 60 GB. For more information about adjusting the size of the PVC, see Increasing Kafka PVC.

Note: For production deployments, the default Kafka PVC is 30 GB, but a minimum of 90 GB is recommended. This adjustment must be made after installing the cluster.

- If available within the console for the connection you are configuring, you can reduce the length of time for historical data collection, reduce the base parallelism, or select a smaller span value in order to reduce the resource usage of the connection.

Examples of performance considerations when collecting data:

- A Dynatrace connector taking too long to collect historical data, with unstable and inconsistent time frames, for example taking 4 days to collect 5 minutes of data.
- An AWS CloudWatch connector collecting more data than the capacity of the metric anomaly detection of the system, for example collecting 100,000 KPIs per 5 minutes while the cluster's sizing supports only 30,000 KPIs per 5 minutes with only 5 metrics selected. The solution to this would be to scale the sizing of the cluster to accomodate more metric data.
- A Zabbix connector pod restarts on a small Red Hat OpenShift Container Platform environment collecting historical metric data and shows an "Unknown" status. The Kafka PVC on this cluster has only 30 GB of storage on a production environment and it is at maximum capacity and the iaf-kafka pod is in a restarting status as well. The solution to this would be to increase the size of the Kafka PVC per the preceding recommendations and to restart the connector-bridge, iaf-kafka, and connector pods.

Planning data loading, checking, and training

You can load and validate data for testing for AI training and operational purposes.

Loading AI training data

When you are loading AI training data, such as to correlate events from various sources, or find anomalies in logs, use the following end-to-end guidance for best results.

1. For each AI type, load *all* relevant historical data. For example, if you are training the log anomaly AI type, first load *all* relevant historical log data by using one or more of the following data connections. Set the connection mode to **Historical data for initial AI training** at the end of the page where you define the data connection.

    - Custom
    - ELK
    - Falcon LogScale
    - Kafka
    - LogDNA
    - Splunk
2. Before you run the training, you can optionally run a data quality precheck on your loaded data to ensure that it is sufficient enough to train on the algorithm selected. Even if you skip this step to run the model training, the data quality precheck still runs as a prerequisite to the model training.
3. Train the AI type, as described in Configuring AI training, and use *all* of the data that you loaded in Step 1. For example, if you are training the log anomaly AI type then follow the instructions to create a log anomaly training definition.
4. Deploy the model for the AI type that you trained in Step 3. Deployment can be automatic or manual.
5. For log anomaly training only, after the model deploys, go back to each data connection that you defined in step 1. Next, change the connection mode to **Live data for continuous AI training and anomaly detection**.

Loading operational data

When you are loading operation data, such as for real-time analysis, use the guidance within the following documentation:

- Incoming connections

  Incoming connections enable IBM Cloud Pak for AIOps to collect log, metric, and event data from various sources.

- Connecting probes and gateways

  If you want to load event data, you can also configure probes and gateways form various sources.

Prerequisites for loading data from domain-specific log data sources

If you configure a log connection to load data from domain-specific log data sources, such as IBM MQ or WebSphere, you must meet specific prerequisites that are different from other log systems.

Domain-specific logs must be sourced from a log management system. You must create an incoming log data connection to collect this data. The type of connection to create depends on the following factors:

- Whether you prefer to provide the data in pull or push mode
- Which log management system stores the domain-specific log data

Modes

Consider the following issues when you are deciding whether to provide the data in pull or push mode.

- Pull
    - Positives:
        - If your log management system has a REST API then all you must do is to create the data connection and Cloud Pak for AIOps does the rest.
        - You can also use the Cloud Pak for AIOps built-in connectors, which can cover commonly used options.

    - Negatives: you have no control over data load. When Cloud Pak for AIOps initiates the pull operation, the operation might create excessive load on your log management systems.
- Push
    - Positives: you have full control over data load.
    - Negatives: you must create code to push your log data to the Kafka topic.

Pull mode

For log data from systems that are not Falcon Logscale, LogDNA, or Splunk, you can make use of either the Custom, ELK, or Kafka data connection to ingest your log data.

Pull mode connections

Domain-specific log management system	Data connection	Link
Falcon LogScale	Falcon LogScale	Falcon LogScale connections
LogDNA	LogDNA	LogDNA connections
Splunk	Splunk	Splunk connections
Log management system that uses an ELK stack	ELK	Custom connections
Any other log management system	Custom	Custom connections

Push mode

For log data from systems that are not Falcon Logscale, LogDNA, or Splunk, you can use either the Custom, ELK, or Kafka data connection to ingest your log data.

Push mode connections

Domain-specific log management system	Data connection	Link
Any system	Kafka	Custom connections

Other requirements

Log entries must be in JSON format, and not in plain text.

When you are preparing domain-specific log data for loading, ensure that the **`ibm_messageId`** and **`loglevel`** fields are at the top level. The **`module`** field is optional. These fields are used by the statistical baseline log anomaly detection algorithm to perform statistical analysis and detect log anomalies.

Deduplication

Occurrences of the same event are deduplicated into a single alert instance.

The default correlation policy constructs a deduplication key for events from the following elements:

```
{{ event.resource }}-{{ event.type.classification }}-{{ event.type.condition }}
```

When the same event occurs multiple times (the resource, classification, and condition fields are the same), then deduplication increments the event count of the current alert instance to show how many times the event occurred.

Problem events (**`type.eventType = problem`**) create an alert (if one does not exist). Meanwhile, resolution events (**`type.eventType = resolution`**) do not become alerts and are not visible in the Alert Viewer. They clear an alert if one exists and ignore if none exists. For a resolution event to clear a problem alert instance, it must have the same deduplication key as the problem event that the alert was created from. Also, it must have occurred after the last problem event that contributed to the alert.

Type fields and alert correlation

Since the same alert arises for all events with same type fields, these fields must clearly indicate the specific fault condition and have separate values for different conditions. The resource fields are important for both alert correlation and topology lookups. Therefore, they must accurately describe the resource that exhibits the fault condition. By default, alerts with the same **`resource.name`** field that occur within 15 minutes of each other are correlated together.

Additionally, all resource fields are used to look up each alert in the topology. Therefore, where possible one or more of these fields must be in the same form as the match tokens for the corresponding topology resource.

Alerts are also correlated by using temporal correlation. Temporal correlation looks for historic patterns of co-occurrence and uses this information to predict future correlation. This correlation relies on consistent value of the type fields over time for the same conditions against the same resources. Therefore, the values of these fields must be deterministic.

HTTP header credential types

The following table shows the different HTTP headers for each credential type.

Note: Different connectors use different credentials, which are outlined in each of the relevant connector topics.

Credential type headers

Credential Type	HTTP Header
API Key	Authorization: **`api-key-value-pair`** For example, X-API-Key: **`abcdef12345`**
API Token	Authorization: Bearer **`token`**
UserID/Password	Authorization: Basic **`username:password`**

Note: The final value of **`username:password`** in the Basic header is a base64-encoded value.

Remote deployment of connectors

You can deploy a connector locally or remotely. Local deployment option deploys the connection in the same cluster where IBM Cloud Pak for AIOps is installed. Alternatively, the remote option deploys it in your chosen location, for example, on a different network region, on Saas, or remotely on premises (VM, cluster, or container).

You can choose the remote option for connector deployment if the connector source is not reachable from the IBM Cloud Pak for AIOps cluster. On selecting the remote option, you will be provided with a bootstrap script after you add the connection to finish the deployment. Run this script on the remote server where you want to deploy the connector.

The bootstrap command creates a secure tunnel between the IBM Cloud Pak for AIOps cluster and the remote cluster on which you are running the connector source. This tunnel feeds the data from the remote server to your IBM Cloud Pak for AIOps cluster.

Prerequisites

Before you run the bootstrap script on a remote cluster, complete the following tasks:

- Install Podman on the remote cluster.
- Log in to Image registry for the script to pull in the connector image by using the Podman login command.

Note: You do not need Red Hat OpenShift running on the remote cluster for deploying the connection remotely.

Procedure

To deploy a connector remotely, complete the following steps:

1. Log in to IBM Cloud Pak Automation console.
2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.

3. On the *Data and tool connections* page, click **Add connection**.
4. From the list of available connections, find and click the tile for the connector that you want to deploy remotely.

   Note: If you do not immediately see the connection that you want to create, you can filter the tiles by type of connection. Click the type of connection that you want in the *Category* section.
5. On the side-panel, review the instructions and when ready to continue, click **Connect**.
6. On the *Add connection* page, go to the *Deployment options* and select **Remote**.
7. Continue with procedure for creating the connector.
8. After the connection is created, select the created connection on the *Data and tool connections* page.
9. Download the bootstrap script by clicking **Download** from the options menu (three vertical dots) for the connection. Copy the script or download the code as an `sh` file.
10. Run the script on the remote cluster where you want to deploy the connector. When the script completes running, the connector is successfully deployed on the remote server.

Defining observer jobs for application and topology data

You can configure observer jobs for collecting application and topology data by using the Observer Configuration UI.

Note: The restriction on the number of concurrent observer jobs has been lifted. Any jobs that are started via the UI or the Observer Service Swagger page that exceed the set threshold of concurrently running jobs (ten) will not be discarded, but instead queued and run as soon as previous jobs complete. Should a scheduled job be prevented from running in such a scenario, it will not fail, though it might run later than scheduled.

Before you begin

For more information on configuring certificates and passwords, see the Configuring observer jobs security topic.

Note: The following observers are enabled by default:

- ServiceNow observer*
- Kubernetes observer*
- VMware vCenter observer*
- File observer
- REST observer

*If you have any restrictive EgressNetworkPolicies in place when connecting to these observer target systems, ensure that they are updated to allow for this inbound connection. For more information, see Postinstallation tasks.

To add other observers

1. In the Operator Lifecycle Management console, go to **Operators > Installed Operators**.
2. Choose the project where IBM Cloud Pak® for AIOps is installed from the **Project** drop-down list.
3. Click the IBM Netcool Agile Service Manager operator.
4. Navigate to the Agile Service Manager tab and select the existing Agile Service Manager instance (such as aiops-topology).
5. Click the YAML tab and edit the file. To enable an observer, set it to `true`, then click **Save**. (**Tip:** To disable it, set it to `false`.)

To enable observers

1. Change to the namespace (project) where AIOps was installed:

   ```
   oc project
   ```

2. Find the ASM release name using the following command:

   ```
   oc get asm
   ```

   Example:

   ```
   oc get asm aiops-topology
   ```

   ```
   NAME             AGE
   aiops-topology   20h
   ```

3. Edit `aiops-topology`:

   ```
   oc edit asm aiops-topology
   ```

   Change the value to `true` for the observers that you want to enable. For example:

   ```
   asm:
     almObserver: false
     ansibleawxObserver: true
     appdynamicsObserver: true
     awsObserver: true
     azureObserver: true
   ```

4. To check all observers have been enabled run the following command:

   ```
   oc get pod | grep aiops-topology
   ```

Tips:

- To disable an observer, set the associated observer parameter to `false` in the ASM release name.
- To verify that the ASM release name has the status of `OK` after the change:

  ```
  oc get asm aiops-topology --no-headers -o custom-columns=":status.phase"
  ```

- To verify that the ASMformation release name has the status of `OK`:

  ```
  oc get asmformation aiops-topology --no-headers -o custom-columns=":status.phase"
  ```

- To manually delete the kubernetes deployment for that observer to free up resources:

  ```
  oc delete deployment -l app.kubernetes.io/name=<observer_name>
  ```

Example

```
oc delete deployment -l app.kubernetes.io/name=alm-observer
```

```
deployment.apps "aiops-topology-alm-observer" deleted
```

To access the Observer Configuration UI

1. Expand **Define**, click **Data and tool connections**, and then click **Add connections**.
2. From the **Category** list (on the left) choose **Topology**, and then click the **Configure, schedule, and manage other observer jobs** link. The Observer jobs page is displayed.
3. Here, you can add a new job, or view, search for, or sort existing jobs by job name, observer name, job status or job type.

Observer jobs

The Observer jobs panel displays all existing jobs in table format. From here, you can interact with existing jobs, or create new ones.

If you have a large number of jobs configured, you can use **Search** to find it.

All existing jobs are listed in a table by name, observer name, job status or job type columns, which are sortable. For each job its state is indicated. Hover over the state to see a tooltip with more information.

If a job has been scheduled, you can hover over the Schedule icon to view more details. The Schedule icon is either a clock icon if scheduled to run at set intervals, or an hourglass icon if it is scheduled to run only once at a future time. When you

cancel a scheduled job (Load jobs only) using **Remove Schedule**, this reverts the job to a default, on-demand job.

You can run a job manually, or switch it On or Off, depending on the job type.

You can also use the **List of options** drop-down to **View & Edit** or **Delete** a job, or **View History** for it (View History is available for Load jobs only).

When you **Delete** a running job, it will not complete its current run, but any previously loaded data will remain. You cannot delete a job that is in the 'Finishing' state.

When you select **View History** for a job, the Observer job history window opens listing the last 10 jobs, when they were run, their observation count, duration, and status. You can refine the list using the **All**, **Complete**, or **Incomplete** tabs.

Click the **Add a new job +** button to open the *Add a new job* panel, which displays all jobs that can be configured in tile format. Click a specific observer tile to open its job configuration window. The Observer job configuration UI lists each job parameter that you must configure.

New 'Generate debug support file' parameter: Set the optional *Generate debug support file* parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer (from **`/var/log/itsm/<obs>-observer`**). **Note** that this functionality is available only via the Observer job configuration UI, and not the Connector UI.

Tip: If an observer stops or is removed, you cannot run existing jobs, or add new jobs. Stopped or removed observers and jobs that are listed in the Observer Configuration UI will be disabled or removed in progressive (housekeeping) steps. If you are reinstalling or reactivating an observer, the jobs and the observer will again become available.

1. **Up to 5 minutes** after removal, observers and jobs still appear as normal until the housekeeping process runs, but cannot be used.
2. **Up to 60 minutes** after removal, the observer is still listed, but jobs are unavailable and marked offline until the next housekeeping process runs. You can delete existing jobs, but cannot view, add or edit jobs.
3. **After 60 minutes** the removed observer is no longer listed, **but jobs remain**, though they are disabled and marked offline. You can delete existing jobs, but cannot view, add or edit jobs.
4. If **at any time** you reinstall or reactivate the observer, it reappears in the UI, and existing (previously active) jobs are no longer disable. You can delete, view or edit existing jobs, or add new jobs.

Configuring ALM Observer jobs

You can define jobs that dynamically load data associated with intent for analysis.

Configuring Ansible AWX Observer jobs

Using the Ansible AWX Observer, you can define load jobs that will gather and read data from a defined Ansible AWX instance. When a load job is run, it will load a baseline topology data view from Ansible AWX through its REST APIs.

Configuring AppDynamics Observer jobs

Using the AppDynamics Observer, you can define a full load job that will read data from the AppDynamics Controller through the REST API. This job can provide, for example, business applications, nodes, and tiers.

Configuring AWS Observer jobs

Using the AWS Observer, you can define jobs that read services data from the Amazon Web Services (AWS) through AWS SDK and generate a topology. It is installed as part of the core installation procedure.

Configuring Azure Observer jobs

Using the Azure Observer, you can define a full load job that will read data from Azure cloud services through its REST APIs and generate a topology.

Configuring Big Cloud Fabric Observer jobs

Using the Big Cloud Fabric Observer, you can define a full load job that will load data from the Big Cloud Fabric REST interface..

Configuring BigFix Inventory Observer jobs

You configure BigFix Inventory Observer jobs to read data from a BigFix Inventory instance through its REST APIs, and generate a topology.

Configuring Ciena Blue Planet Observer jobs

Using the Ciena Blue Planet Observer, you can define jobs that will gather and read all topology data from the Blue Planet MCP instance by REST API and generate a topology.

Configuring Cisco ACI Observer jobs

You use the Cisco ACI Observer when you have a Cisco ACI environment with Cisco Application Policy Infrastructure Controller (APIC) in your environment. The Observer interfaces with Cisco APIC and makes active REST calls to Cisco APIC in the Cisco ACI environment. You configure observer jobs that dynamically load Cisco ACI data for analysis from the Observer Configuration UI.

Configuring Contrail Observer jobs

Using the Contrail Observer, you can retrieve topology data from Juniper Network Contrail Release 4.1 through REST APIs exposed by the Contrail apiserver. This observer is developed against Juniper Network Contrail that integrates with OpenStack orchestration platform (Ubuntu 18.04 + Contrail Cloud - Ocata).

Configuring DNS Observer jobs

Using the DNS Observer, you can query internal DNS server performance, and use the returned information on response times and service addresses to create topologies within the topology service. The DNS Observer supports forward and reverse job types, with 'recurse' or 'no recurse' options.

Configuring Docker Observer jobs

Using the Docker Observer, you can discover Docker network resources, including Docker Swarm clusters, and then visualize (or model) this data as a topology view in the UI. You configure observer jobs from the Observer Configuration UI.

Configuring Dynatrace Observer jobs

Using the Dynatrace Observer, you can query a specified Dynatrace environment for information about its applications, services, process groups, and infrastructure entities.

Configuring File Observer jobs

Using the File Observer functionality, you can write bespoke data to a file in a specific format. You can upload this data to the topology service, and then visualize this data as a topology view in the UI.

Configuring GitLab Observer jobs

Using the GitLab Observer functionality, you dynamically load GitLab projects and their child resources to the topology service, and then visualize this data as a topology view in the UI.

Configuring GoogleCloud Observer jobs

Using the GoogleCloud Observer, you can define a full load job that will read services data from the Google Cloud Platform's Compute Services through Google's Compute Services SDK, and then generate a topology.

Configuring HP NFVD Observer jobs

Using the HP NFVD Observer, you can dynamically load data from the HP NFVD REST interface.

Configuring IBM Cloud Observer jobs

Use the IBM Cloud Observer when you install IBM Cloud in your environment to run jobs that read data from an IBM cloud instance. These jobs retrieve Cloud Foundry Apps information and services, and then dynamically load the retrieved data for analysis.

Configuring Jenkins Observer jobs

Using Jenkins Observer, you can define listen jobs that receive build information generated by the plug-in for Jenkins.

Configuring Juniper CSO Observer jobs

Using the Juniper CSO Observer, you can define a full load job that will gather and read data about topology data from Juniper CSO. The observer is installed as part of the core installation procedure.

Configuring Kubernetes Observer jobs

Using this observer, you can configure jobs that discover the structure of your Kubernetes clusters, including pods, worker nodes and containers.

Configuring Network Manager Observer jobs

Using the IBM Tivoli Network Manager Observer, you can define jobs that dynamically load data discovered by Network Manager for analysis.

Configuring New Relic Observer jobs

Use New Relic Observer when you have a New Relic account with a New Relic Infrastructure subscription. Using New Relic Observer, you can configure jobs that dynamically load New Relic Infrastructure resource data through New Relic for analysis.

Configuring OpenStack Observer jobs

Using the OpenStack Observer, you can configure jobs that dynamically load OpenStack data for analysis.

Configuring Rancher Observer jobs

Using the Rancher Observer, you can define a full load job that will read data from Rancher through the REST API and generate a topology.

Configuring REST Observer jobs

Use the REST Observer for obtaining topology data through REST endpoints. This observer is a counterpart to the File Observer.

Configuring SDC ONAP Observer jobs

You configure SDC (Service Design and Creation) ONAP (Open Network Automation Platform) Observer jobs to dynamically load data from the SDC REST interface.

Configuring ServiceNow Observer jobs

Using the ServiceNow Observer job, you can retrieve the configuration management database (CMDB) data from ServiceNow through REST API, using basic authentication credentials.

Configuring SevOne Observer jobs

Using the SevOne Observer functionality, you can load monitored servers and their relationships, and then visualize this data as a topology view.

Configuring Tivoli Application Dependency Discovery Manager Observer jobs

IBM Tivoli Application Dependency Discovery Manager Observer jobs retrieve network topology data, including discovered applications, their components, configurations and dependencies, from the Tivoli Application Dependency Discovery Manager database server (running either an IBM Db2 or an Oracle database), and use this data to create topology views.

Configuring Viptela Observer jobs

You configure Viptela Observer jobs to load managed Viptela SD-WAN control plane, vEdge and their associated network resources.

Configuring VMware NSX Observer jobs

You configure VMware NSX Observer jobs to dynamically load data from the VMware NSX REST interface.

Configuring VMware vCenter Observer jobs

You configure VMware vCenter Observer jobs to dynamically load data from the VMware vCenter REST interface.

Configuring Zabbix Observer jobs

Using the Zabbix Observer functionality, you can load monitored servers and their associated network resources, and then visualize this data as a topology view in the UI. It is installed as part of the core installation procedure.

Configuring ALM Observer jobs

Using the Agile LifeCycle Manager Observer, you can define jobs that dynamically load data associated with intent from the Agile LifeCycle Manager for analysis.

Before you start

Important: The ALM Observer supports IBM Agile Lifecycle Manager 2.2.

Attention: The ALM observer supports the discovery of Cloud Pak for Network Automation (CP4NA) and has been tested against versions 2.3.x and 2.4.x. The ALM observer was first developed to integrate with the legacy Telco Network Cloud Manager (TNCM) Orchestrator (TNCO/ALM). A re-branding from ALM (Agile Lifecycle Manager) to CP4NA happened, while the observer retained its ALM name. The observer has some level of backward compatibility with the legacy ALM.

Ensure you have the Agile LifeCycle Manager Kafka server host and topics to hand, such as the Agile Lifecycle Manager server, the Kafka port, and the topics used for lifecycle events.

Important: To access Agile LifeCycle Manager remotely, you must ensure that the Agile LifeCycle Manager installation is configured with the KAFKA_ADVERTISED_HOST_NAME so as to allow remote connections. For more information, see the Configuration reference topic in the Agile LifeCycle Manager documentation: Configuration reference

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The Agile LifeCycle Manager Observer jobs listen to the Kafka 'state change' topics of Agile Lifecycle Manager, as well as the Agile Lifecycle Manager Resource Manager. Information is extracted from Agile Lifecycle Manager about Assemblies and Resources and a topology is created.

After installation, you define and start the following two jobs.

Listen for Agile Lifecycle Manager events ('alm' job)

The **alm** job is a long-running job that monitors its source for updates and runs until it is explicitly stopped, or until the observer is stopped.

Listen for Resource Manager events ('rm' job)

The **rm** job is a long-running job that monitors its source for updates and runs until it is explicitly stopped, or until the observer is stopped.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Important: The value of the IBM Agile Lifecycle Manager instance name parameter needs to be the same for both jobs to allow the topology to be combined.

Listen for lifecycle events ('alm' job) parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required

Parameter	Action	Details
IBM Agile LifeCycle Manager instance name	Use this to identify the Agile LifeCycle Manager installation and any associated Resource Managers	Required
Group ID	Use this to identify the Kafka group ID to be used	Required
Connection	Use this to specify the Kafka Host and Port to be used	Required
ALM State Change Topic	Use this to identify the Agile LifeCycle Manager Kafka topic name to be consumed.	Required
ALM Service State Transition Topic	Use this to identify the Agile LifeCycle Manager State Transition Kafka topic name to be consumed.	Required
Secured or unsecured Kafka connection	Select false or true to make secured or unsecured connection to the CP4NA server. The default is 'false'.	Required. When set to 'true', you must also provide the credentials and target system certificate parameters. See the following topic for more information on obtaining connection and credential details: IBM Cloud Pak for Network Automation: Configuring Kafka connections
Kafka username	Enter the username of the secured Kafka cluster.	Required only if the parameter for secured connection has been set to 'true'.
Kafka password	Enter the password of the secured Kafka cluster.	Required only if the parameter for secured connection has been set to 'true'.
Certificate	Enter the certificate of the secured Kafka cluster.	Required only if the parameter for secured connection has been set to 'true'. For more information, see Configuring observer jobs security
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional

Listen for Resource Manager events ('rm' job) parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
IBM Agile Lifecycle Manager instance name	Use this to identify the Agile LifeCycle Manager installation and any associated Resource Managers	Required
Group ID	Use this to identify the Kafka group ID to be used	Required
Connection	Use this to specify the Kafka Host and Port to be used	Required
Topic	Use this to identify the Agile Lifecycle Manager **Resource Manager** Kafka topic	Required

Parameter	Action	Details
Secured or unsecured Kafka connection	Select false or true to make secured or unsecured connection to the CP4NA server. The default is 'false'.	Required. When set to 'true', you must also provide the credentials and target system certificate parameters. See the following topic for more information on obtaining connection and credential details: IBM Cloud Pak for Network Automation: Configuring Kafka connections
Kafka username	Enter the username of the secured Kafka cluster.	Required only if the parameter for secured connection has been set to 'true'.
Kafka password	Enter the password of the secured Kafka cluster.	Required only if the parameter for secured connection has been set to 'true'.
Certificate	Enter the certificate of the secured Kafka cluster.	Required only if the parameter for secured connection has been set to 'true'. For more information, see Configuring observer jobs security
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional

Configuring Ansible AWX Observer jobs

Using the Ansible AWX Observer, you can define load jobs that will gather and read data from a defined Ansible AWX instance. When a load job is run, it will load a baseline topology data view from Ansible AWX through its REST APIs.

Before you begin

Important: The Ansible AWX Observer supports the on-premises Ansible AWX version 21.14.0 and Tower API version 3.8.6.

Ensure you have the Ansible AWX details to hand, such as the host, port, user and token before defining and running the observer job.

Required: Before defining the observer load job, you must obtain a personal access token (PAT) from AWX. The token authentication is recommended for the usage of Tower's API.

1. Access the AWX UI.
2. From the menu bar, navigate to the **Users** page that is under the **Access** section.
3. Select a username, then click the **Token** option.
4. Click the **+** (plus) icon.
5. Leave the application field blank, and choose **read** for the token's access.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

You define and start the following job.

Full Topology Upload job (through restapi)

By default, Load jobs are one-off, transient jobs that upload all of the requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job.	Required
Ansible AWX hostname	Specify the Ansible AWX hostname	Required
Ansible AWX host port	Specify the Ansible AWX host port	Required
Username (owner) of the personal access token (PAT)	Specify the username (owner) of the personal access token (PAT)	Required
Personal access token (PAT)	Specify the personal access token (PAT)	Required.
HTTPS truststore file name	Specify the truststore file name. The observer generates the trust store file based on the file name provided. **Tip:** You can use the observer name as file name (`<observer>.jks`), for example `Ansible.jks`.	Required. The supported format is JKS and the file is relative to $ASM_HOME/security.
truststore file password	Specify the truststore password the observer will use to decrypt the HTTPS truststore file. Use a password that conforms to your internal security requirements.	Required
Target system certificate	Specify a certificate by name to automatically create and load into the truststore	Required
SSL verification	Choose whether SSL validation is true or false. Turning SSL validation to `false` will use HTTPS without host verification.	Optional
Number of jobs to display for each template	Specify the number of recent jobs that will be modeled on the topology for each of the job template. Max value is 50. Larger value might result in the topology being less perceptible	Optional
Connection timeout (milliseconds)	Enter a connection timeout in milliseconds	Optional. Must be a value greater than 0 (zero), and the default is `5000` (5 seconds).
Read timeout (milliseconds)	Enter a read timeout in milliseconds	Optional
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional

Parameter	Action	Details
Job schedule	Specify when the job runs.	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job.	Optional

Configuring AppDynamics Observer jobs

Using the AppDynamics Observer, you can define a full load job that will read data from the AppDynamics Controller through the REST API. This job can provide, for example, business applications, nodes, and tiers.

Before you start

Important: The AppDynamics Observer supports the cloud and SaaS AppDynamics version 4.5.12 and API version 4.5.x.

Ensure you have the AppDynamics details to hand, such as the instance, account, username and password before running the observer job.

Tip: Before defining the observer load job, you must create an AppDynamics user with the correct permissions. This is required for REST API authentication.

1. On the AppDynamics Administration page, click the gear icon (upper right).
2. Under the Admin group, select **Administration**, then select the **Users** tab.
3. Select the **AppDynamics** option from the **Display users** drop-down list.
4. Assign all available roles to the user by selecting **Add from Roles** > **Select All** > **Done**, then click **Save**.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The AppDynamics Observer imports ITSM Resource Topology Service data.

You define and start the following job.

Full Topology Upload job

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Instance	Specify the AppDynamics Controller instance	Required
Account	Specify the Tenant account name	Required
Username	Specify the name of the user of the specified tenant account	Required
Password	Specify the password for the specified user	Required

Parameter	Action	Details
SSL Verification	Choose to use SSL validation or not. The options are `true` or `false`.	Optional. The default is 'false', which uses HTTPS validation without hostname validation.
Connection timeout	Specify the connection timeout in milliseconds.	Optional. Default is 5000 (5 seconds).
Read timeout	Specify the read timeout in milliseconds.	Optional. Default is 5000 (5 seconds).
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Configuring AWS Observer jobs

Using the AWS Observer, you can define jobs that read services data from the Amazon Web Services (AWS) through AWS SDK and generate a topology. The AWS Observer supports multiple Amazon web services such as EC2 for its 'elastic compute' services.

Before you begin

Important: The AWS Observer supports the cloud and SaaS AWS version 1.11.

The AWS Observer supports EC2 service and currently processes EC2 resources in the following order. When the account in use has insufficient privileges to access a resource, the observer skips the resource(s) and proceed to the next one.

- Instance
- AvailabilityZones
- NetworkInterfaces
- Volumes
- SecurityGroups
- VPCs
- Subnets
- RouteTables

Important: As a minimum, the account in use must have the DescribeInstances policy, or the observer job fails. For more information, see Identity and access management for Amazon EC2.

About this task

You define and start the following job.

Full Topology Upload job

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Data Tenant	Specify the unique data tenant name to track the discovered data for each job.	Required
Access Key	Specify the AWS access key	Required
Secret Key	Specify the AWS secret key	Required
Region	Specify the AWS region or multiple regions to discover	Required
Property to exclude	Single or multiple properties to exclude	Optional
Proxy	Specify the proxy (`true` or `false`)	Required. If set to `true`, you must supply the proxy host, port and protocol.
Proxy Host	Specify the proxy host	Optional (unless proxy is set to `true`)
Proxy Port	Specify the proxy port	Optional (unless proxy is set to `true`)
Proxy Protocol	Specify the proxy protocol	Optional (unless proxy is set to `true`)
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Configuring Azure Observer jobs

Using the Azure Observer, you can define a full load job that will read data from Azure cloud services through its REST APIs and generate a topology.

Before you begin

Important: The Azure Observer supports the cloud and SaaS Azure version.

Ensure you have the Azure details to hand, such as the Tenant ID, Client ID, and client password before running the observer job.

Tip: To configure access to Azure resources, see the following information: Create an Azure Active Directory application and service principal that can access resources

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The Azure Observer retrieves topology data from Azure cloud services through REST APIs exposed by the Azure API server.

In addition to the standard resources, the following resources are now also supported:

- Load Balancers
- Virtual Machine Scale Set
- Virtual Network Gateway (including VPN Gateway and Express Route Gateway)
- ExpressRoute

You define and start the following job.

Full Topology Upload job

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Data center	Enter the data center of the Azure service	Required
Tenant ID	Specify the Azure account tenant ID	Required
Client ID	Specify the client ID to access the registered app	Required
Client password	Enter the client password to access the registered app	Required
Subscriptions to be discovered	Specify the subscription Ids to be discovered in comma-separated lists. To discover all subscriptions, use `*`	Optional
Connection timeout (milliseconds)	Define the Azure connection timeout	Optional
Proxy Host	Specify the proxy host	Optional
Proxy Port	Specify the proxy port; defaults to 8080	Optional
Connection timeout (milliseconds)	Define the Azure connection timeout in milliseconds. The default value is 5000.	Optional
Read timeout (milliseconds)	Specify the read timeout in milliseconds. The default value is 5000.	Optional
Proxy Username	Specify the proxy username.	Optional
Trust all certificates by bypassing certificate verification	Set to true to allow connection to target environment without verification.	Optional. The default is 'false'.
Proxy Password	Specify the proxy password.	Optional
Proxy Secure	Specify whether the proxy server is secure.	Optional

Parameter	Action	Details
Access Scope	Optional CSV String listing values which can be used to provide a scope for the resources. These can be used to aid the mapping of alerts to resources when resources in different scopes share the same matchTokens. Example of scope include locations, project names and namespaces.	Optional
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Configuring Big Cloud Fabric Observer jobs

You configure Big Cloud Fabric Observer jobs to dynamically load data from the Big Cloud Fabric REST interface.

Before you begin

Ensure you have the Big Cloud Fabric service details to hand, such as username, password, hostname and port.

About this task

The Big Cloud Fabric Observer job extracts Big Cloud Fabric resource information through REST and loads and updates the resources and their relationships within the topology service.

You define and start the following job:

LDAP job

Run this job whenever you need Big Cloud Fabric topology data refreshed.

By default, these jobs are one-off, transient jobs that perform a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) manually from the Observer UI, or schedule them to run at set times when configuring them.

The Big Cloud Fabric Observer loads the following resources and their relationship:

Big Cloud Fabric resources and relationships loaded by the Big Cloud Fabric Observer LDAP job

Vertex	Description
Access Leaf	Access Leaf Switch part of DC fabric
Access Port	Access Leaf Switch port connected to Compute Node and Spine Switch port
Spine Switch	Spine Switch part of DC fabric
Spine Switch Port	Spine Switch port connected to Access Leaf Switch and L3 Router Port

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

LDAP job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Proxy hostname	Specify the proxy server name	Required
Proxy Port	Specify the proxy port number for the REST API	Required
Proxy username	Specify the proxy username	Required
Proxy password	Specify the proxy password	Required. Plain text password can be entered.
BCF authentication URL	Specify the BCF authentication URL.	Required
BCF Controller Host Name	Specify the BCF controller hostname in string format. The hostname is taken from the job parameter.	Required. **Note:** The BCF controller hostname value (together with the proxy hostname and username) identifies the source of data for a Big Cloud Fabric Observer job instance. If your BCF controller hostname changes after a job is created, you must create more than one job instance for each BCF controller hostname. If the BCF controller hostname **does not** change after the job is created, then create a single job instance with all BCF controller hostnames specified.
HTTPS or HTTP connection to the BCF authentication server	Specify whether the connection to the BCF authentication server should be HTTPS (secured) or HTTP.	Required
HTTPS or HTTP connection to the BCF proxy server	Specify whether the connection to the BCF proxy server should be HTTPS (secured) or HTTP.	Required
Trust all certificates Verification	Choose whether Trust all certificates Verification is true or false.	Required. When set to true, the observer will trust all certificates.
Delay between two api calls (in seconds)	Specify the delay between retries	Optional
Concurrency Level	Specify the Concurrency Level, with a minimum of 3 and a maximum of 5.	Optional
Authentication system certificate	Specify the BCF Authentication system certificate to be used (if any).	Optional. Required if the HTTPS or HTTP connection to the BCF authentication server parameter is true. To obtain authentication certificates using OpenSSL and store them as secrets, see the Configuring observer jobs security topic.
Proxy system certificate	Specify the BCF proxy system certificate to be used (if any)	Optional. Required if the HTTPS or HTTP connection to the BCF proxy server parameter is true. To obtain proxy certificates using OpenSSL and store them as secrets, see the Configuring observer jobs security topic.
SSL Host Name Verification	Choose whether SSL validation is true or false.	Optional. When SSL validation is set to false, the observer will make HTTPS connections without validating the proxy hostname of the provided certificate.
Connection timeout (milliseconds)	Enter the time at which the connection actions time out.	Optional. Must be a value greater than 0 (zero), and the default is 5000 (5 seconds).
Read timeout (milliseconds)	Enter the time at which the read actions time out.	Optional. Must be a value greater than 0 (zero), and the default is 5000 (5 seconds).

Parameter	Action	Details
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job should run, and whether it should run at regular intervals. By default the job runs immediately, and only once. Optionally you can specify a future date and time for the job to run, and then set it to run at regular intervals after that.	Optional. Transient (one-off) jobs only. If you set a job schedule, the run intervals must be at least 90 seconds apart, and if you set them at less than 15 minutes, a warning is displayed, as the frequency can impact system performance.
Observer job description	Enter additional information to describe the job	Optional

Configuring BigFix Inventory Observer jobs

You configure BigFix Inventory Observer jobs to read data from a BigFix Inventory instance through its REST APIs, and generate a topology.

Before you begin

Important: The BigFix Inventory Observer supports BigFix Inventory 9.5.0.

Before configuring a BigFix Inventory job, ensure you have the BigFix Inventory details to hand such as the BigFix Inventory URL, API token and SSL truststore.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

You define and start the following job.

BigFix Inventory Observer job (full topology load)

Run this job whenever you need BigFix Inventory topology data refreshed.

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required

Parameter	Action	Details
BigFix Inventory API token	Enter the BigFix token for authentication	Required. Must be encrypted.
BigFix Inventory instance URL	Specify the API URL of the BigFix Inventory endpoint (including port)	Required. Usually in the following format: https://<hostname or IP address>:<port>
BigFix Inventory resources	Specify the resources to be discovered	Optional. Lists supported values such as software, hardware or `*`. If left blank, all available resources are discovered.
BigFix Inventory certificate	Specify the name of the certificate to be loaded into the trust store	Required
HTTPS truststore file name	Specify the truststore file name	Required
truststore file password	Specify the truststore password to decrypt the HTTPS truststore file	Required
SSL Validation	Choose whether SSL validation is on or off. Turning off SSL validation will use HTTPS without host verification.	Optional
BigFix Inventory connection timeout (ms)	Enter the time at which the connection times out	Optional. Must be a value greater than 0 (zero), and the default is 5000 (5 seconds).
Data Center	Specify the data centers in which the BigFix Inventory instance runs	Required.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Troubleshooting: If you configure a BigFix Inventory Observer job, the pod for the job might not start. After you configure the job, if you view the list of all observers you might not see the BigFix Inventory Observer job (**`bigfixinventory-observer`**) pod listed with a status of running.

If you encounter this issue, you can manually enable the job by changing the value of **`asm.bigfixinventoryObserver.enabled`** to true:

```
helmValuesASM:
  asm.bigfixinventoryObserver.enabled: true
```

To change this value, you can run the following command:

```
oc patch asm.asm.ibm.com/aiops-topology --type=json -p
'[{"op":"add","path":"/spec/helmValuesASM/asm.bigfixinventoryObserver.enabled","value":true}]'
```

To verify that the pod is now running, run the following command:

```
oc get pods | grep -i big
aiops-topology-bigfixinventory-observer-7f8c465698-gtm5p          1/1     Running
0          5m
```

Configuring Ciena Blue Planet Observer jobs

Using the Ciena Blue Planet Observer, you can define jobs that will gather and read all topology data from the Blue Planet MCP instance by REST API and generate a topology.

Before you begin

Important: The Ciena Blue Planet Observer supports MCP 4.0.

Ensure you have the Ciena Blue Planet details to hand, such as API username, API password, MCP URL, MCP certificate, truststore file and truststore password.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The Ciena Blue Planet Observer has two jobs, the restapi load and WebSocket listen jobs.

- When a **load** job is run, it loads baseline topology data through Blue Planet MCP APIs: Network Elements (constructs), EquipmentHolder, Equipment, TPE (Terminating Point Encapsulation), and FRE (Forwarding Relationship Encapsulation).
- When a **listening** job is run, the observer listens to changes in resource's status from the BluePlanet MCP instance through a WebSocket connection.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Full Topology Upload job (through restapi)

By default, Load jobs are one-off, transient jobs that upload all of the requested topology data as soon as they are triggered.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Server URL	The URL of the MCP server instance	Required
MCP Certificate	Certificate name to load into the truststore	Required
SSL truststore file	Exact HTTPS truststore file name	Required
SSL truststore password	The password to decrypt HTTPS truststore file	Required
SSL Validation	Choose whether SSL validation is on or off. Turning off SSL validation will use HTTPS without host verification.	Optional
Connection timeout	Sets the connection and read timeout in milliseconds	Optional. Must be a value greater than 0 (zero), and the default is 5000 (5 seconds).
Username	MCP API username	Required
Password	MCP API password	Required
Data Center	The data center the MCP instance is running in	Required
Tenant Name	The tenant to use	Required
Job schedule	Specify when the job runs	Optional. Load jobs only.

Parameter	Action	Details
Observer job description	Enter additional information to describe the job	Optional

WebSocket Listen job parameters

A long-running job that monitors its source for updates and runs until it is explicitly stopped, or until the observer is stopped.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
WebSocket URL	The MCP WebSocket URL to connect to	Required
WebSocket Topics Subscriptions	The MCP topics to subscribe to	Required
Server URL	The URL of the MCP server instance	Required
MCP Certificate	Certificate name to load into the truststore	Required
SSL truststore file	Exact HTTPS truststore file name	Required.
SSL truststore password	The password to decrypt HTTPS truststore file	Required
SSL Validation	Choose whether SSL validation is on or off. Turning SSL validation off will use HTTPS without host verification.	Optional
Connection timeout	Sets the connection and read timeout in milliseconds	Optional. Must be a value greater than 0 (zero), and the default is 5000 (5 seconds).
Username	MCP API username	Required
Password	MCP API password	Required
Data Center	The data center the MCP instance is running in	Required
Tenant Name	The tenant to use	Required
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional

Configuring Cisco ACI Observer jobs

You use the Cisco ACI Observer when you have a Cisco ACI environment with Cisco Application Policy Infrastructure Controller (APIC) in your environment. The Observer interfaces with Cisco APIC and makes active REST calls to Cisco APIC in the Cisco ACI environment.

Before you begin

Important: The Cisco ACI Observer supports the on-premises Cisco ACI version 4.1.

Ensure you have the Cisco ACI service details to hand, such as the Cisco APIC username, Cisco APIC password, Cisco APIC SSL truststore and Cisco APIC URL.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

Optional: Install and configure a proxy

The Cisco ACI Observer can (optionally) connect to Cisco ACI through a proxy. See the Proxy Host and Proxy Port parameters listed in the table. A proxy can be used to create a tunnel to the target system.

Example: Squid proxy

1. Install the squid proxy:

   ```
   sudo yum install squid
   ```

2. Edit the /etc/squid/squid.conf file to add ACL for the source and target system. Example of a line added:

   ```
   acl localnet src 1.2.3.0/19 # ASM Host
   ```

3. Initialize the squid directories:

   ```
   sudo squid -z
   ```

4. Configure the squid proxy for auto-start:

   ```
   sudo systemctl start squid
   sudo systemctl enable squid
   ```

For more information about the squid proxy, see http://www.squid-cache.org.

About this task

A Cisco ACI Observer job extracts Cisco ACI resources from Cisco APIC through REST, and then the observer loads and updates the resources and their relationships within the topology view.

You define and start the following jobs.

Restapi Load job

A transient (one-off) job that loads all requested topology data using Cisco APIC REST APIs to build a tenant logical construct topology or a fabric topology, and then exits.

A 'restapi' job loads initial topology data, and can resynchronize topology data from Cisco ACI when required.

You assign 'restapi' as the job type for /jobs/restapi observer endpoint.

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

WebSocket Listen job

A long-running job that listens for notifications from Cisco APIC to build the topology and runs until it is explicitly stopped, or until the observer is stopped.

A 'websocket' job monitors changes from Cisco APIC object notification and updates the topology.

You always run a 'websocket' job **after** running a 'restapi' job type.

You assign 'websocket' as the job type for /jobs/websocket observer endpoint.

Cisco ACI objects and relationships loaded

The Cisco ACI Observer loads the following Cisco ACI objects and their relationships into the topology service:

Tenant Logical construct

- fvTenant
- fvAp - A policy owner in the virtual fabric

- fvAEPg - A set of requirements for the application-level EPG instance
- fvAEpP - Abstract representation of an endpoint profile
- fvEpP - An endpoint profile
- fvBD - A bridge domain is a unique layer 2 forwarding domain that contains one or more subnets
- fvCtx - The private layer 3 network context that belongs to a specific tenant or is shared
- vzBrCP - A contract is a logical container for the subjects, which relate to the filters that govern the rules for communication between endpoint groups (EPGs)
- vzOOBBrCP - An out-of-band binary contract profile can only be provided by an out-of-band endpoint group and can only be used by the external prefix set
- vzSubj - A subject is a subapplication running behind an endpoint group (for example, an Exchange server). A subject is parented by the contract, which can encapsulate multiple subjects
- vzFilter - A filter policy is a group of resolvable filter entries
- fvSubnet - A subnet defines the IP address range that can be used within the bridge domain
- fvRsCons -The Consumer contract profile information and on Cisco ACI console the option to create this object is through Consumed Contract. Used to build relationship between fvAEPg and vzBrCP
- fvRsBd - A source relation to the bridge domain associated to this endpoint group. Used to build relationship between fvBD and fvAEPg
- fvRsCtx - A source relation to a private layer 3 network context that either belongs to a specific tenant or is shared. Used to build relationship between fvBD and fvCtx
- vzRsSubjFiltAtt - The filter for the subject of a service contract. Used to build relationship between vzSubj and vzFilter

Fabric Topology

- fabricInst - A container object for fabric policies
- fabricNode - The root node for the APIC
- polUni - Represents policy definition or resolution universe
- firmwareRunning - Information about leaf or spine switch firmware running on a node
- firmwareCtrlrRunning - Information about each controller firmware that is running
- eqptLCSlot - The slot for the module card
- eqptLC - A line card (IO card) contains IO ports
- eqptPsuSlot - The power supply slot
- eqptPsu - The power supply unit
- eqptFtSlot - A fan tray slot
- eqptFan - The fan in a fan tray
- topSystem - Used to retrieve fabric node Operational State
- cnwPhysIf - The physical interface that is assigned to the node cluster
- l1PhysIf - The object that represents the Layer 1 physical Ethernet interface information object
- mgmtMgmtIf - The management interface
- lldpAdjEp - The LLDP neighbors, which contains the information regarding the neighbors
- eqptRsIoPhysConf - A source relation to an L1 Ethernet interface. Used to build relationship between l1PhysIf and eqptLC
- mgmtRsOoBStNode - An object that contains management ip address of fabric spine switches and fabric leaf switches

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: For more information, see the Configuring observer jobs security topic.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Tenant name	Use this to identify the tenant	Required. Set to `admin` if no specific tenant is referenced. Set to '' to load Fabric Topology resources.

Parameter	Action	Details
Cisco APIC endpoint	Specify the API URL of the Cisco APIC endpoint	Required. Usually in the following format: https://[hostname or IP address]/api
Cisco APIC username	Specify the username to connect as, or listen to	Required
Cisco APIC password	Enter the password for Cisco APIC authentication	Required
HTTPS truststore file name	Specify the truststore file name	Required
HTTPS truststore file password	Specify the truststore password to decrypt the HTTPS truststore file	Required
Cisco APIC certificate	Specify a certificate by name to load into the truststore	Required.
Read timeout	Specify the read timeout in ms (default is 2000)	Optional
SSL Validation	Choose whether SSL validation is on or off. Turning SSL validation off will use HTTPS without host verification.	Optional
Connection timeout	Specify the connection timeout in ms (default is 5000)	Optional
Proxy Host	Specify the proxy host through which to connect	Optional
Proxy Port	Specify the proxy port, defaults to 8080	Optional
Trust all certificates by bypassing certificate verification	Set to true to allow connection to target environment without verification.	Optional. The default is 'false'.
Access Scope	Optional CSV String listing values which can be used to provide a scope for the resources. These can be used to aid the mapping of alerts to resources when resources in different scopes share the same matchTokens. Example of scope include locations, project names and namespaces.	Optional
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Configuring Contrail Observer jobs

Using the Contrail Observer, you can retrieve topology data from Juniper Network Contrail Release 4.1 through REST APIs exposed by the Contrail apiserver. This observer is developed against Juniper Network Contrail that integrates with OpenStack orchestration platform (Ubuntu 18.04 + Contrail Cloud - Ocata).

Before you begin

Important: The Contrail Observer supports the on-premises Contrail version 4.1.0.

Ensure you have the Contrail API Server and OpenStack credentials details to hand. For rabbitmq jobs, you must also specify the location of the RabbitMQ queue and its authentication details.

About this task

You define and start the following jobs.

Rabbitmq Listen job

A long-running job that monitors its source for updates and runs until it is explicitly stopped, or until the Observer is stopped.

This job loads all supported resources during startup, and listens to RabbitMQ messages from 'vnc_config.object-update' fanout exchange.

The restapi job does not have to be run before running the rabbitmq job because the rabbitmq job performs a restapi job during initialization.

Restapi Load job

This job loads all supported resources.

Run this job whenever you need the Contrail topology data refreshed.

By default, Load jobs are one-off, transient jobs that upload all of the requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Rabbitmq Listen job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
RabbitMQ hostname or IP address	Enter a hostname or IP address for the RabbitMQ server	Required
RabbitMQ port	Specify the port for connection	Required
RabbitMQ username	Specify the username for authentication with RabbitMQ	Required
RabbitMQ password	Enter the password for authentication with RabbitMQ	Required
RabbitMQ virtual host	Enter the RabbitMQ virtual hostname	Optional. Default is /.
Contrail API URL	Specify the URL for the Contrail apiserver	Required
OpenStack Authentication URL	Enter the authentication URL for the identity service	Required
OpenStack username	Specify the OpenStack username to connect as (or to)	Required
OpenStack password	Specify the OpenStack password with which to authenticate	Required

Parameter	Action	Details
Authentication type	Specify the authentication type used.	Optional. The default is **Keystone**. The other option is **None**.
OpenStack project domain name	Specify the OpenStack project domain name	Optional
OpenStack user domain name	Specify the OpenStack project user domain name	Optional
OpenStack project name	Enter the OpenStack project name for version 3 authentication	Optional
OpenStack tenant name	Enter the OpenStack tenant name for version 2 authentication	Optional
OpenStack identity API version	Select an option from the drop-down list	Optional
Connection and read timeout (milliseconds)	Choose the time at which the connection and read action times out	Optional. Must be a value greater than 0 (zero). The default is 5000 (5 seconds).
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional

Restapi Load job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Contrail API URL	Specify the URL for the Contrail API server	Required
OpenStack Authentication URL	Enter the authentication URL for the identity service	Required
OpenStack username	Specify the OpenStack username to connect as (or to)	Required
OpenStack password	Specify the OpenStack password with which to authenticate	Required
Authentication type	Specify the authentication type used	Optional. The default is **Keystone**, the other option is **None**.
OpenStack project domain name	Specify the OpenStack project domain name	Optional
OpenStack user domain name	Specify the OpenStack project user domain name	Optional
OpenStack project name	Enter the OpenStack project name for version 3 authentication	Optional

Parameter	Action	Details
OpenStack tenant name	Enter the OpenStack tenant name for version 2 authentication	Optional
OpenStack identity API version	Select an option from the drop-down list	Optional
Connection and read timeout (milliseconds)	Choose the time at which the connection and read action times out	Optional. Must be a value greater than 0 (zero). The default is 5000 (5 seconds).
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Configuring Datadog Observer jobs

Using the Datadog Observer functionality, you can load monitored hosts and their associated processes, and then visualize this data as a topology view in the IBM Cloud Pak for AIOps UI.

BETA The Datadog Observer is available only for technology preview, and is not for production usage.

Before you start

Important: The Datadog Observer supports the cloud/SaaS Datadog version.

Ensure you have the Datadog server details to hand, such as the data tenant, API server URL, Auth API key and App key.

About this task

A Datadog Observer job extracts host and its associated processes from Datadog via REST API. The Observer loads and updates the resources and their relationships within the IBM Cloud Pak for AIOps core topology service.

You define and start the following job.

Full Topology Upload job

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Table. Parameter details

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Data Tenant	Enter the data tenant to track the Datadog topology	Required
API server URL	Enter the Datadog API base URL	Required
Auth API key	Specify the authorized API key to authorize with Datadog	Required
App key	Specify the application key to access the Datadog resources	Required
Configure Host API query	Specify the host API query. Each query parameters needs to be separated by ";" and its value needs to be inside "()". For parameters that support list, the values need to be separated by "," and API will treat them as AND operation	Optional
Configure Process API query	Specify the process API query. Each query parameters needs to be separated by ";" and its value needs to be inside "()". For parameters that support list, the values need to be separated by "," and API will treat them as AND operation	Optional
Datadog certificate	Specify the Datadog certificate to enable secure connection to the target system	Optional. For more information, see Configuring observer jobs security.
Certificate validation	Choose whether to use SSL validation ('true' or 'false'). Turning SSL validation off will use HTTPS without host verification.	Optional
Connection and read timeout (milliseconds)	Set the connection and read timeout value.	Optional
Trust all certificates	Trust all certificates without validating the certificate's contents. Choose 'true' or 'false'.	Optional
Number of connection retries	Set the number of times a connection should try again. The default value is 3.	Optional
Delay before a connection retry (milliseconds)	Specify the period of time in milliseconds to delay before a connection tries again.	Optional
Access Scope	Optional CSV String listing values which can be used to provide a scope for the resources. These can be used to aid the mapping of alerts to resources when resources in different scopes share the same matchTokens. Example of scope would be locations, project names, namespaces, etc.	Optional
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Configuring DNS Observer jobs

Using the DNS Observer, you can query internal DNS server performance, and use the returned information on response times and service addresses to create topologies within the topology service. The DNS Observer supports forward and reverse job types, with 'recurse' or 'no recurse' options.

Before you begin

Ensure you have the DNS access details to hand, such as DNS server, address types and port numbers.

Tip: The relationship types can be customized with line color, width and pattern functions. For more information, see the Creating custom relationship type styles topic.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The DNS Observer provides DNS query services and topological insight into how a specified DNS server is performing forward (name-to-IP address) or reverse (IP address-to-name) lookups. Query results include a list of addresses, information on how long it takes the DNS server to resolve a lookup, and, optionally (with the maximum number of recursive calls set at 200) how the DNS server is recursively resolving a given name or IP address.

Note: The DNS observer models the resources based on the DNS lookup record. A separate set of resources will be created for IPv4 and IPv6 discovery of the same domain name.

You define and start the following jobs.

Reverse lookup job

A transient (one-off) Load job that loads all requested DNS reverse (IP address-to-name) lookup topology data.

By default, Load jobs are one-off, transient jobs that upload all of the requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Forward lookup job

A transient (one-off) Load job that loads all requested DNS forward (name-to-IP address) lookup topology data.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: For more information, see the Configuring observer jobs security topic.

Reverse lookup job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
ip_address	Enter the host or internet name to look up	Required
Address Types	Specify the address types to be observed	Required. Select either IPv4 or IPv6.
Server	Specify the DNS server	Required
Port	Specify the UDP DNS port	Optional
Recursive query	Toggle **True** or **False**	Optional. If set to True, the maximum number of calls is set at 200.

Parameter	Action	Details
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Forward lookup job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Domain name	Specify a domain	Required
Address Types	Specify the address types to be observed	Required. Select either IPv4 or IPv6.
Server	Specify the DNS server	Required
Port	Specify the UDP DNS port	Optional
Recursive query	Toggle **True** or **False**	Optional. If set to True, the maximum number of calls is set at 200.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Configuring Docker Observer jobs

Using the Docker Observer, you can discover Docker network resources, including Docker Swarm clusters, and then visualize (or model) this data as a topology view.

Before you begin

Important: The Docker Observer supports Docker 3.1.0.

Note: Docker UCP 3.1.0 supports only TLS 1.2 for SSL negotiation and no longer supports TLS 1 and TLS 1.1.

Ensure you have the details for your Docker job to hand, specifically your Docker system's Unix socket, and or host and port number.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

Using the Observer Configuration UI you configure observer jobs that query the Docker REST API to retrieve data and display it as a topology in the Topology Viewer. The Docker Observer can model external Docker systems, and it can also provide a System health view of the Docker system on which IBM Cloud Pak® for AIOps runs.

The job parameters connect to a remote Docker using the host and port parameters.

Remote Docker

The 'host' and 'port' parameters of the job can be used to identify the TCP port that Docker can be reached on. The **unix_socket** parameter must not be supplied.

Docker is not accessible through TCP by default. To enable it, edit the 'docker.service' file. On Red Hat® OpenShift® Container Platform, this is available in `/usr/lib/systemd/system`. Amend the **ExecStart** option under the **Service** section to include a `-H` option. For example, to make it available externally on port 2375, you might add `-H tcp://0.0.0.0:2375`.

Note: If you want to continue to be able to access Docker through the default socket, for example if the Docker Observer container needs access, or if you want to be able to perform `docker ps -a` rather than `docker -H tcp://0.0.0.0:2375 ps -a`, then you need to also list it in the same line, as on the following example:

```
-H tcp://0.0.0.0:2375 -H unix:///var/run/docker.sockCopy
```

You must reload the configuration:

```
sudo systemctl daemon-reload
sudo systemctl restart docker
```

Tip: If this fails to start Docker, and a Unix socket (or no socket at all) was specified, check that no directory with that name exists. If you start Docker with just a TCP socket and no Unix socket, this creates a `/var/run/docker.sock` directory, which you must delete after Docker is stopped, so that you can restart with access through that Unix socket.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Host	Use this to identify the TCP host socket (HTTP or HTTPS) on which to access the remote Docker system	Required for **remote** Docker access only
Username	Specify the username of the remote Docker environment with HTTPS	Required for remote Docker with HTTPS access only
Password	Specify the password of the remote Docker environment with HTTPS	Required for remote Docker with HTTPS access only.
Docker SSL Certificate	Specify the certificate file name	Optional
Docker SSL truststore File	Specify the truststore file name	Required for remote Docker with HTTPs access only.
SSL truststore File Password	Specify the truststore password	Required for remote Docker with HTTPs access only.
Port	Use this to identify the TCP port (HTTP or HTTPS) on which to access the remote Docker system	Required for **remote** Docker access only
Unix Socket	Use this to access local docker environments using the complete path	Required for **local** Docker access only. Host and port parameters **must** be empty.

Parameter	Action	Details
View	Use this to select which resources are modeled in the topology view	Optional. The Default displays running resources only. Options are **Container** (all running containers), **Image** (images used by running containers), and **Task** (running tasks only)
Containers to exclude	List container you want to exclude	Optional
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Configuring Dynatrace Observer jobs

Using the Dynatrace Observer, you can query a specified Dynatrace environment for information about its applications, services, process groups, and infrastructure entities.

Before you begin

Important: The Dynatrace Observer supports Dynatrace cloud, SaaS and Dynatrace onPrem v1.182.102.

Ensure you generate a Dynatrace token to access your Dynatrace environment. You also need topology access scope to access the Dynatrace resources.

Ensure you have the Dynatrace access details to hand, such as Dynatrace API URL and API token.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

You define and start the following job.

Dynatrace Load job

A transient (one-off) job that loads all requested Dynatrace resource data.

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Notice: Dynatrace v1 API endpoints are no longer supported.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Dynatrace Observer job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Data tenant	Data tenant is used to track your data. It needs to be unique for each job or else it will override the previously discovered data.	Required
Base URL	Specify the API URL of the Dynatrace endpoint (including version). An example of a Dynatrace **cloud-based URL** is: `https://< your-environment-id >.live.dynatrace.com/api/v2/`, while an example of a Dynatrace managed or **onPrem URL** is: `https://< your-domain >/e/< your-environment-id >/api/v2/`	Required
Dynatrace API Token	Enter the Dynatrace token for authentication	Required
Enable/Disable Secure Connection to CSO's host	Select to either enable or disable a secure connection to the CSO's host. Set to true to secure the connection, or false to bypass it.	Required
The entitySelectors for the Dynatrace v2 API	The format is `{"taskname" : "entitySelector"}`, and multiple tasks can be added to the entitySelector list. The **taskname (string, required)** is a user defined value that identifies each task uniquely. The **entitySelector** defines the scope of the query. Only entities matching the scope and criteria will be discovered. Separate each filter of entitySelector with ';' and place its value inside parentheses '( )'. For a filter that supports a list, separate the values using a comma ','. For more *entitySelector information, see more details and example below this table.	Required
Number of records to be fetched per page	Number of records to be fetched per page.	Optional. Default value is 500.
Relationship mapping for dynatrace with asm	Observer relationship to be used for Dynatrace relationship. The user can use edge mapping values to also override the default values	Optional. Observer will use a default mapping for edges.
Entity mapping for dynatrace with ASM	Observer entity types to be used for Dynatrace entity type. Observer will use a default mapping for entity. The user can use entity mapping values to also override the default values.	Optional. If there is no default value then Dynatrace entity type will be used as ASM entity type.
Grouping vertex preference	The observer creates the defined topology for the discovered vertices based on the preference in this field.	Optional. Value can be **host** or **none**. - Default value is **none**, which indicates that no resource groups are created. - When **host** is selected, resource groups are created with the host as `seed` and its connected entities, except other hosts, as members.
Dynatrace certificate	Specify a certificate if secure connection to CSO's host is true.	Required if secure connection to CSO's host is enabled.
Certificate validation	Option for SSL validation. Turning SSL validation off will use HTTPS without host verification.	Optional

Parameter	Action	Details
Connection and read timeout (milliseconds)	Set the connection and read timeout value.	Optional. Default value is 5000.
Trust all certificates	Trust all certificates without validating the certificate's contents.	Optional. Default value is true.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs.	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job.	Optional

Note: For the **Grouping vertex preference** job parameter, when **host** is selected for creating resource groups, the host unique ID is set as the seed during the processing of the host API response. Connected entities from the response (excluding connected hosts) are used as the members of the created resource group. In Dynatrace, entities like application, service, process and even custom data can be part of several hosts.

Dynatrace API entitySelector parameters and examples:

Examples:

Single task example:

```
{
"HostTask": "type(host);entityId(HOST-E9B8C2EF21BA5217,HOST-B343896535B69B84,HOST-
F4A80CBC1027416A);timeframe(now-10m);fields(properties.osType,properties.hypervisorType)"
}
```

Multiple tasks example:

```
{
"HostTask":"type(host);timeframe(now-10m);fields(properties)",
"AppTask":"type(application);timeframe(now-10m);fields(properties)",
"ServiceTask":"type(service_instance);timeframe(now-10m);fields(properties)",
"ProcessTask":"type(process_group_instance);timeframe(now-10m);fields(properties)"
}
```

Parameters:

- **!** (negation): For example `!entityName(eur01.ibm.com)`
- **type (string, required)**: Each task can only accept one type. You must set this criteria in the entitySelector. Negation is not supported by type.
- **mzName (string, list, optional)**: Management zone name of the entity
- **mzId (string, list, optional)**: Management zone id of the entity
- **entityName (string, list, optional)**: Name of the entity
- ✱ (wildcards): For example the string `"type(host);entityName(eur0*);..."` will fetch all of the hosts that start with an entityName of 'eur0', such as 'eur01.ibm.com,eur02.ibm.com', etc; while the string `"type(host);entityName(*eur0*);..."` will fetch all hosts that contain 'eur0' as part of their entityName parameters, such as `ibm.eur0.com,ibm.eur02.com`, etc.
- **entityId (string, list, optional)**: Id of the entity
- **tag (string, list, optional)**: Tags of the entity
- **state (string, list, optional, case-sensitive):** State of the entity
- **timeframe (string, optional)**: Time frame of the entity. The format is `now-NU`, where 'N' is the amount of time, and 'U' is the unit of time. The default is '(now-1h)'. Supported time units for the relative time frame are 'm' (minutes), 'h' (hours), 'd' (days), 'w' (weeks), 'M' (months), and 'y' (years).
- **fields (string, list, optional)**: Entity fields that will be added to the entity vertex

Configuring File Observer jobs

Using the File Observer functionality, you can write bespoke data to a file in a specific format, upload this data to the topology service, and then visualize this data as a topology view.

Before you begin

The File Observer reads topology data from specified files, and uploads it. You must create these files manually.

About this task

Topology data in a file is composed of vertices (nodes) and edges. A vertex represents an object (resource), while an edge represents the relationship between two objects.

Each line of the file you create should be in one of the following formats, loading a single resource vertex (including optional relationships in the '_references' field) or a single edge, deleting a single vertex, or pausing execution.

Lines starting with **`V:`** (vertex), **`E:`** (edge), **`D:`** (delete) or **`W:`** (wait) are treated as instruction lines to be processed. Other lines, for example lines that are empty or commented out, are ignored.

Line format

V - Load a resource vertex, with a JSON representation as documented for the body of the topology service API method: **`POST /resources`**

If specifying the '_status' element, acceptable state values are open, closed, or clear, and acceptable severity values are clear, indeterminate, warning, minor, major, or critical.

E - Load an edge, with a JSON representation as documented for the '_references' section of the body of the topology service API method **`POST /resources`**

D - Delete a resource vertex, identified by it's **`uniqueId`**.

W - Pause for the duration (for testing purposes only). Takes an integer period followed by a string specifying the units.

G - To load a group via the File Observer, see the following example syntax:

```
G:{"uniqueId":"EA","entityTypes":["group"],"name":"Event Analytics"}
```

Note: When creating a group's edge, it must be based on **`uniqueId`**.

Important: Ensure that the file is structured correctly. For each line of the file, information included after the closing **}** that matches an opening **{** is ignored, and no error is recorded.

Mapping alerts to topologies When using the File Observer to correlate events from Logs Anomaly, ensure you use the **`matchTokens`** property when creating topology templates. The **`matchTokens`** property is a list of values you can use to look up a resource from an alert. If a unique **`matchToken`** is found that matches an alert's resource fields, such as **`resource.name`** or **`resource.sourceId`**, the alert will be assigned to that resource.

File Observer syntax

When uploaded to the file observer in a file, say **`example.txt`**, the following small topology can be generated by running a File Observer job for that file. The figure also shows an alert matched to a resource.

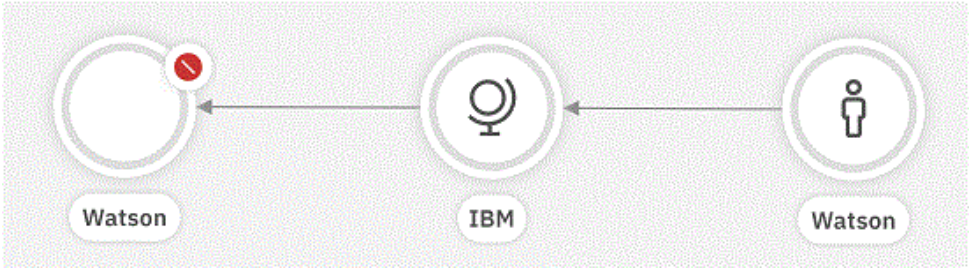


Figure. File Observer topology

```
V:{"uniqueId":"Thomas J. Watson","entityTypes":["person"],"name":"Watson"}
V:{"uniqueId":"IBM","entityTypes":["company","domain"],"name":"IBM","_references":
[{"_edgeType":"owns","_toUniqueId":"Watson AIOps"},{"_edgeType":"contributedTo",
"_fromUniqueId":"Thomas J. Watson"}]}
V:{"uniqueId":"Watson AIOps","entityTypes":
["ai"],"name":"Watson","platform":"OCP","matchTokens":["AIOps","Watson AIOps"]}
```

Line 1: This line has the minimal set of properties for a resource vertex:

- **`uniqueId`** - Uniquely identifies the resource within the provider namespace (the provider is supplied when running the job).
- **`entityTypes`** - Determines the icon displayed in the UI. Some types have a pre-defined mapping to icons, as described in the Entity types topic.
- **`name`** - The label displayed in the UI.

Line 2: This line shows multiple entityTypes. There is no default icon mapped to type company, so the icon is determined from the domain type. In addition to the properties of the resource, this adds an edge to another resource.

- The edgeType must exist already. Either one of the default Edge types or one created by the topology service's **`POST /types/edge`** API (see Accessing APIs)
- The referenced uniqueId does not have to exist yet.
- Multiple edges can be added in the **`_references`** list.

Line 3: This line shows that more data or properties can be added. It adds a custom **`platform`** property (might be any additional property required). See the Properties topic for a list of existing properties. This line also illustrates that any **`entityType`** can be supplied, and that the **`name`** does not have to be unique. The UI does not show an icon for this resource because no default icon mapping has been created. You can add custom mapping through the UI as described in the Defining custom icons topic. And finally, this line also adds a **`matchToken`** property, which is a list of values that can be used to link a resource to an alert.

File Load job

By default, Load jobs are one-off, transient jobs that upload all of the requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Save** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required.
File Mode	Select either **Upload local file**, then drag and drop (or browse) to the input file; or **Use server file**, then specify the name of the file previously uploaded through the **`POST /files`** API.	Required. **Note:** If the same file name is used in a second job, or the file is replaced through **`PUT /files`**, any resources missing from the second load will be deleted.

Parameter	Action	Details
Provider	Enter the job's provider, that is, a namespace for the resources' uniqueIds. Resources can be loaded from multiple files with the same provider, and linked as if they were all loaded from the same file. To link to resources from another provider (for example, from another observer), the cross-provider semantics are required.	Optional. Defaults to `FILE.OBSERVER:{fileName}` if not supplied.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Swagger API documentation

For the Swagger API documentation, see Topology File Observer API. The documentation explains how to use the File Observer APIs.

File Observer API access

Two public APIs are available, and both APIs can be used to retrieve data.

- **API name**

  topology-file-observer

- **Public API URLs**

  `/aiops/api/application-manager/topology-file-observer/v1/*`

  `/aiops/api/app/topology-file-observer/v1/*`

For information on how to obtain route and token information, see Accessing the console.

For information on how to obtain your token, see Accessing APIs.

Example API queries

GET /health API

```
curl -k -X GET --header 'Accept: application/json' -H "Authorization: Bearer ${TOKEN}"
https://$ROUTE/aiops/api/application-manager/topology-file-observer/v1/healthcheck
{"date":"2021-02-05T19:24:00.974+00:00","jobs":0,"threads":10,"items":[{"type":"load"}]}
```

```
curl -k -X GET --header 'Accept: application/json' -H "Authorization: Bearer ${TOKEN}"
https://$ROUTE/aiops/api/app/topology-file-observer/v1/healthcheck
{"date":"2021-02-05T19:24:00.974+00:00","jobs":0,"threads":10,"items":[{"type":"load"}]}
```

GET /service/info API

```
curl -k -X GET --header 'Accept: application/json' -H "Authorization: Bearer ${TOKEN}"
"https://$ROUTE/aiops/api/application-manager/topology-file-observer/v1/service/info"
{"startDate":"2021-02-04T10:30:56.991+00:00","id":"example-noi-topology-file-observer-
6f88fb6fc4-n9xm2","name":"file-observer","version":"1.1.0.17
66dd27a","adminConnector":"https://example-noi-topology-file-observer-6f88fb6fc4-
n9xm2:9099","appConnector":"https://example-noi-topology-file-observer-6f88fb6fc4-
n9xm2:9098","uptimeMillis":118575702}
```

```
curl -k -X GET --header 'Accept: application/json' -H "Authorization: Bearer ${TOKEN}"
"https://$ROUTE/aiops/api/app/topology-file-observer/v1/service/info"
{"startDate":"2021-02-04T10:30:56.991+00:00","id":"example-noi-topology-file-observer-
6f88fb6fc4-n9xm2","name":"file-observer","version":"1.1.0.17
66dd27a","adminConnector":"https://example-noi-topology-file-observer-6f88fb6fc4-
n9xm2:9099","appConnector":"https://example-noi-topology-file-observer-6f88fb6fc4-
n9xm2:9098","uptimeMillis":118575702}
```

Upload a file

```
curl -k -X POST --header 'Accept: application/json' -H "Authorization: Bearer ${TOKEN}"
https://$ROUTE/aiops/api/app/topology-file-observer/v1/files --form
'job_file=@/tmp/file.json.txt' --header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255'
```

Creating cross-data origin relationships

Using the File Observer functionality, you can create edges between resources from different observations or providers.

It requires both the **uniqueID** and the provider of the resources at each end of the link (so typically applies to previously observed resources).

In the following syntax examples, note the **_toProvider** and **_fromProvider** edge properties.

You can define an edge from a resource in the current observation to one owned by a different provider:

```
V:{"uniqueId":"host1.ibm.com","entityTypes":["host"],"name":"host1","_references":
[{"_edgeType":"connectedTo","_toProvider":"someOtherProvider","_toUniqueId":"host2.ibm.com"}]}
```

You can define an edge from a resource between two different providers:

```
E:{"_fromProvider":"providerOne","_fromUniqueId":"host1-
interface1","_edgeType":"connectedTo","_toProvider":"providerTwo","_toUniqueId":"host2-
interface1"}
```

File Observer sample files

You can upload sample file data to the topology service on File Observer, and then view this data in the topology. To download the sample files, go to Sample files.

Configuring GoogleCloud Observer jobs

Using the GoogleCloud Observer, you can define a full load job that will read services data from the Google Cloud Platform's Compute Services through Google's Compute Services SDK, and then generate a topology.

Before you begin

Important: The Google Cloud Observer supports the cloud and SaaS Google Cloud version.

The GoogleCloud Observer supports GoogleCloud's compute services. Ensure you have the GoogleCloud details in hand, such as the Project ID, Service Account Key File and Zone, before running the observer job.

The GoogleCloud Observer supports a transient (one-off) Load job that loads all requested topology data through Google's Compute Services SDK to build the topology, and then exits.

To create a service account key file

1. From the Google Cloud Platform dashboard, under your 'Project ID', go to APIs and Services and then choose **Credentials**. The Credentials page is displayed listing a number of authentication methods.
2. Select the **Service account** authentication service
3. From **Create Credentials**, choose **Service account**.
4. Complete the service account details, then click **Create** to create a new service account.

5. Select the **Compute Engine > Compute Admin** role, then click **Continue** and **Done**.
6. To download the JSON file from the Credentials page, click **Service account > Manage Service Accounts**.
7. Select the new service account, then choose **Create key** from the overflow menu (the three dots under the **Actions** column).
8. Choose a key type of JSON, then click **Create**. A JSON file will be downloaded. For more information on storing the service account key file as a secret, see the Configuring observer jobs security topic. The file name will be used in the observer parameter (service_account_key_file) for the full load job.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

You define and start the following job. You must edit the parameters in the configuration file before running this job.

Full Topology Upload job

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) manually from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: For more information, see the Configuring observer jobs security topic.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Project ID	Enter the Google Cloud Platform Project ID	Required
Service Account Key File	Supply the Google Cloud Platform Service Account Key File	Required
Zone	Specify the Google Cloud Platform Zones	Required.
Connection timeout	Specify the connection timeout in ms (default is 5000)	Optional
Read timeout (milliseconds)	Specify the read timeout in milliseconds. The default value is 5000.	Optional
Proxy Host	Specify the proxy host via which to connect.	Optional
Proxy Port	Specify the proxy port. Set as per HTTP/HTTPS Proxy Host.	Optional. Defaults to 8080.
Proxy Username	Specify the proxy username. Set for basic auth proxy.	Optional
Trust all certificates by bypassing certificate verification	Set to true to allow connection to target environment without verification.	Optional. The default is 'false'.
Proxy Password	Specify the proxy password. Set if Proxy Username has been specified.	Optional. Use plain text.
Proxy Secure	Specify whether the proxy server is secure. Set to 'true' for HTTPS proxy.	Optional. Default is 'false'.
Access Scope	Optional CSV String listing values which can be used to provide a scope for the resources. These can be used to aid the mapping of alerts to resources when resources in different scopes share the same matchTokens. Example of scope include locations, project names and namespaces.	Optional

Parameter	Action	Details
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Note: You must create a service account key file or use an existing one to allow the GoogleCloud Observer to discover resources from GoogleCloud.

Troubleshooting: While the job is running, the status of discovered resources may appear as 'indeterminate' in the topology until the full upload is complete.

Configuring HP NFVD Observer jobs

You configure HP NFVD Observer jobs to dynamically load data from the HP NFVD REST interface.

Before you begin

Important: The HP NFVD Observer supports the HPE NFV Director API release 6.0.3.

Ensure you have the HP NFVD service details to hand, such as username, password, hostname and port.

About this task

The HP NFVD Observer job extracts HP NFVD resource information through REST. The observer loads and updates the resources and their relationships and then displays them as a topology view.

You define and start the following jobs.

Load job

Run this job whenever you need HP NFVD topology data refreshed.

By default, these jobs are one-off, transient jobs that perform a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) manually from the Observer UI, or schedule them to run at set times when configuring them.

Fileload job

Run this job to consumes topology data from a file.

The fileload job loads a baseline topology data view from a provided data file.

The fileload job is a transient (one-off) job that consumes topology data from the file, sends the data for processing, and then stops.

The fileload job provides an alternative to loading topology data through a file.

Listen job

A long-running job that reads LCM events published by HP NFVD on CGMB (Kafka) topic continually until it is explicitly stopped, or until the observer is stopped.

LCM messages in Kafka are typically encoded in Avro and are decoded by the observer using an HP NFVD LCM notification Avro schema file. Refer to the HP NFVD documentation on how to obtain the schema file. The additional steps required when using an Avro schema file are documented in the following subsection of the procedure.

The job should only be run *after* a load job has been completed.

HP NFVD resources and relationships loaded by the HP NFVD Observer job

The HP NFVD Observer loads the following resources and their relationship into the Agile Service Manager core topology service:

Vertex	Description
Organization	Organization (for example, LOB) within Domain
VDC	Tenant within Organization
VNF Group	Group of VNF / CNF
VNF	Virtual Network Function
CNF	Containerized Network Function
VNFM	Manager of VNF/CNF
VM	Virtual Machine attached to VNF
Network Interface / Port	Network Interface or Port attached to VNF
Connection Point	Internal Connection Point
VNF-C	VNF Component
CNF-C	CNF Component
Virtual Link	Virtual Network attached to VNF
Virtual Storage	Storage attached to VNF
Deployment	K8s resources attached to CNF
Statefulset	K8s resources attached to CNF

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: For more information, see the Configuring observer jobs security topic.

Load job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
HP NFVD data center name	The name of the data center on which the HP NFVD is hosted.	Required. **Important:** To observe the complete lifecycle of resources, configure the same data center name, username and hostname across all jobs.
HP NFVD username	Specify the username.	Required. **Important:** To observe the complete lifecycle of resources, configure the same data center name, username and hostname across all jobs.
HP NFVD hostname	Specify the hostname of the HP NFVD.	Required. **Important:** To observe the complete lifecycle of resources, configure the same data center name, username and hostname across all jobs.
HP NFVD password	Enter the password with which to authenticate.	Required. Plain text password can be entered.
HP NFVD host port	Specify the port of the HP NFVD.	Required

Parameter	Action	Details
HTTPS or HTTP connection to the HP NFVD server	Choose to make either a HTTPS (secured) or HTTP connection to the HP NFVD server.	Required. When set to true, provide the target system certificate to the certificate parameter.
True for a CNF job and false for a VNF job	The VNF job is expected to run with fewer threads and will take longer to complete. If this filter parameter is set to `false`, cnf_group_filters defines group names to exclude. If this filter parameter is set to `true`, cnf_group_filters defines group names to include.	Required
Target system certificate	Specify a certificate by name to load into the trustStore.	Required if the `HTTPS connection to the HP NFVD server` parameter is set to true. To obtain authentication certificates using OpenSSL and store them as secrets, see Configuring observer jobs security.
SSL Verification	Choose whether SSL validation is true or false. Turning SSL validation false will use HTTPS without hostname validation.	Optional
Connection and read timeout	Enter the time at which the connection and read actions time out.	Optional. Must be a value greater than 0 (zero). The default is 5000 (5 seconds).
Correlate	Set this to true to enable Event Analytics correlation on the topology grouping based on VNF Groups created by this job.	Optional
CNF group filters	If specified, this parameter selects CNF groups by partial matching on group name. For example, if `CNF1,CNF2` is specified, then a CNF job will only process matching groups such as 'MyCNF1Yeh' and 'YourCNF2No', and a VNF job will process groups that do not match.	Optional
First retry after delay (milliseconds)	Set the initial delay for a retry.	Optional
Max elapsed time for retries (milliseconds)	Set the maximum elapsed time for retries.	Optional
Job schedule	Specify when the job should run, and whether it should run at regular intervals. By default the job runs immediately, and only once.Optionally you can specify a future date and time for the job to run, and then set it to run at regular intervals after that.	Optional. Transient (one-off) jobs only. If you set a job schedule, the run intervals must be at least 90 seconds apart, and if you set them at less than 15 minutes, a warning is displayed, as the frequency can impact system performance.
Observer job description	Enter additional information to describe the job.	Optional

Fileload job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
HP NFVD data center name	The name of the data center on which the HP NFVD is hosted.	Required. **Important:** To observe the complete lifecycle of resources, configure the same data center name, username and hostname across all jobs.
HP NFVD username	Specify the username.	Required. **Important:** To observe the complete lifecycle of resources, configure the same data center name, username and hostname across all jobs.

Parameter	Action	Details
HP NFVD hostname	Specify the hostname of the HP NFVD.	Required. **Important:** To observe the complete lifecycle of resources, configure the same data center name, username and hostname across all jobs.
Name of the VNF or CNF data file	Specify the name of the VNF or CNF data file. Wildcards (*) are supported. Use separate data files for CNF and VNF resource types. **Tip:** If the filename contains the token `nfvd-cnf-data-dump/nfvd_cnf_data_dump`, it is treated as a **CNF** resource. If the filename contains the token `nfvd-vnf-data-dump/nfvd_vnf_data_dump` then it is treated as a **VNF** resource.	Required. The data file must be located in the $/opt/ibm/netcool/asm/data/hpnfvd-observer directory.
HP NFVD archive file retention days	Specify the number of days that the HP NFVD archive files are retained.	Optional. Default is 5 days.
Correlate	Set this to true to enable Event Analytics correlation on the topology grouping based on VNF Groups created by this job.	Optional
CNF group filters	If specified, this parameter selects CNF groups by partial matching on group name. For example, if `CNF1,CNF2` is specified, then a CNF job will only process matching groups such as 'MyCNF1Yeh' and 'YourCNF2No', and a VNF job will process groups that do not match.	Optional
First retry after delay (milliseconds)	Set the initial delay for a retry.	Optional
Max elapsed time for retries (milliseconds)	Set the maximum elapsed time for retries.	Optional
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job should run, and whether it should run at regular intervals. By default the job runs immediately, and only once.Optionally you can specify a future date and time for the job to run, and then set it to run at regular intervals after that.	Optional. Transient (one-off) jobs only. If you set a job schedule, the run intervals must be at least 90 seconds apart, and if you set them at less than 15 minutes, a warning is displayed, as the frequency can impact system performance.
Observer job description	Enter additional information to describe the job.	Optional

Listen job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required

Parameter	Action	Details
HP NFVD data center name	The name of the data center on which the HP NFVD is hosted.	Required. **Important:** To observe the complete lifecycle of resources, configure the same data center name, username and hostname across all jobs.
HP NFVD username	Specify the username.	Required. **Important:** To observe the complete lifecycle of resources, configure the same data center name, username and hostname across all jobs.
HP NFVD Kafka broker host	Specify the hostname of the HP NFVD.	Required. **Important:** To observe the complete lifecycle of resources, configure the same data center name, username and hostname across all jobs.
HP NFVD Kafka broker port	Specify the port number of the Kafka broker.	Required
The HP NFVD Kafka topic to listen for LCM notifications	Specify the topic name for LCM notifications.	Required
True for a CNF job and false for a VNF job	Specify `True` for a CNF job, or `False` for a VNF job. The VNF job is expected to run with fewer threads and will take longer to complete.	Required. **Important:** To observe the complete lifecycle of resources, this must be the same as the option used in the Load job.
SSL	Choose to turn SSL on (or off) for Kafka connection. Turn this option on if the Kafka connection is secured using SSL.	Optional
Kafka TrustStore file	Specify the truststore file used to establish a secured connection to the Kafka broker	Required if SSL has been enabled. For TrustStore and KeyStore configuration, see Avro schema file.
Kafka trustStore password	Enter the password of the Kafka truststore	Required if a Kafka TrustStore file is specified. Use plain text.
Kafka KeyStore file	Specify the keyStore file used to establish a secured connection to the Kafka broker	Required if SSL has been enabled. For TrustStore and KeyStore configuration, see Avro schema file.
Kafka keyStore password	Enter the password of the Kafka keyStore	Required if a Kafka KeyStore file has been specified. Use plain text.

Parameter	Action	Details
Avro Decoding	Choose whether Avro decoding is on or off (True or False). Set to `True` if incoming LCM messages are Avro encoded. If so, the Avro schema file is required.	Optional. Default is false.
Avro schema file	The name of the Avro schema file to be used in decoding binary Avro Kafka messages.	Required if Avro Decoding is set to true. Complete the Avro schema file setup.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job.	Optional

To use the Avro schema file

Avro schema file setup is required if the Avro decoding parameter is set to `True`.

To use the LCM Avro schema file, create a ConfigMap for the schema file to be mounted to the existing volume. If Kafka is in secured mode (SSL), additionally create two ConfigMaps (that is, truststore and keystore to be mounted to the existing volume).

Assumption: The Kafka keystore and truststore files are already set up together with the LCM Avro schema file.

1. Create a configMaps to mount the schema files to the existing volume.

   a. Create ConfigMaps to mount schema files to the existing volume.

   The example below creates the ConfigMap Avro schema file named 'lcm-notification-schema.avsc':

   ```
   oc create configmap hpnfvd-lcm-notification-schema --from-file=lcm-notification-schema.avsc
   ```

   If Kafka is secured (SSL), keystore/truststore is most likely required. The example below creates ConfigMaps using the keystore file named 'keystore', and the truststore file named 'truststore':

   ```
   oc create configmap hpnfvd-kafka-truststore --from-file=truststore
   ```

   ```
   oc create configmap hpnfvd-kafka-keystore --from-file=keystore
   ```

   After creating the ConfigMap, you can verify the results using one of the following commands:

   ```
   oc describe <configMap name>
   ```

   or

   ```
   oc get <configMap name>
   ```

   b. Map the ConfigMap to a volume in the pod's deployment.

   Add to the existing volume, (asm-security), under 'volume Mounts' and 'volumes':

   ```
   volumeMounts:
     - mountPath: /opt/ibm/netcool/asm/security         <--- existing mount
       name: asm-security

   volumes:
     - name: asm-security

     - configMap:
       name: hpnfvd-lcm-notification-schema
       optional: false
       items:
   ```

```
    - key: lcm-notification-schema.avsc
      path: lcm-notification-schema.avsc
```

You can verify the results using the following command:

```
oc exec -it <pod-name> -- ls /opt/ibm/netcool/asm/security
```

The schema files should be listed.

Configuring IBM Cloud Observer jobs

Use the IBM Cloud Observer when you have IBM Cloud installed in your environment to run jobs that read data from an IBM cloud instance. These jobs retrieve Cloud Foundry Apps information and services, and then dynamically load the retrieved data.

Before you begin

Important: The IBM Cloud Observer supports the cloud and SaaS IBM Cloud version.

Important: The IBM Cloud Observer supports Cloud Foundry API version 2.92.

Ensure you have the IBM Cloud access details to hand to specify and access the cloud instance, such as the instance ID, credentials, and region.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

You define and start the following job.

Full Topology Upload job

The IBM Cloud Observer imports ITSM Resource Topology Service data.

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Mapping IBM Cloud model objects to entity types

IBM Cloud resource object	Entity types
stacks	`operatingsystem`
apps	`application`
routes	`path`
service bindings	`hub`
service instance	`service`
user provided service instance	`service`
spaces	`group`
organization	`organization`
buildpacks	`component`

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Instance	Enter the name of the IBM cloud instance.	Required
User email	The email address used to access the instance, or enter `apikey` if IBM Cloud API Key is used.	Required. See Creating an API key on steps to create an IBM Cloud API key.
Password	Enter the password or API key used to access the instance.	Required
IBM Cloud Region	Choose the cloud instance region from the drop-down list (US, UK, EU, AP)	Required. Each region has its own URI, and only a single region is discovered in a full load job. To discover different regions, a full load job needs to be triggered for each region.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs.	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job.	Optional

Configuring Jenkins Observer jobs

Using the Jenkins Observer, you can define listen jobs that receive build information generated by the Jenkins plug-in.

Before you begin

Ensure you have all required Jenkins details to hand.

Before you configure the Jenkins Observer, you must install and configure the Jenkins plug-in provided with the installation images. The plug-in sends your instrumented build data to the Jenkins Observer in order to generate the appropriate topology information: Configuring the Jenkins plug-in.

For the Jenkins plug-in to be able to communicate and send information to a Jenkins Observer job, you must enable a URL route for the Jenkins Observer, which is not exposed in your installation by default.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

You define and start the following job.

Listen job: The stand-alone listen job receives Jenkins build notification data for a specified namespace and processes it as a topology.

The listen job is long-running, and runs until it is explicitly stopped or until the observer is stopped.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Jenkins observation namespace	Specify a provider namespace with which to associate all the topology data generated by this job	Required
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional

Configuring the Jenkins plug-in

The IBM Cloud Pak® for AIOps software includes the Jenkins plug-in, which you install on your Jenkins server using the Jenkins Plugin Manager Advanced installation web UI. From the Jenkins server, the plug-in gathers and sends information to the Jenkins Observer.

Ensure you have all the required administrator or management permissions to deploy the IBM Cloud Pak for AIOps Jenkins component onto the Jenkins server.

Plug-in	Version	Description
Jenkins core	2.159	Jenkins server minimum version
apache-httpcomponents-client-4-api	4.5.10-2.0	Provided by Jenkins core
workflow-step-api	2.22	
workflow-cps	1.9	
workflow-multibranch	2.17	
credentials	2.3.1	Compatible version usually provided by Jenkins core
Git	2.6.5	
artifactory	3.15.0	Optional (for using the Artifactory integration only)

Obtain the Jenkins plug-in

You can find the Jenkins plug-in included with the Jenkins Observer image. For example:

```
JENKINS_OBSERVER_POD=(oc get po -l app.kubernetes.io/name=jenkins-observer -o
jsonpath='{.items[0].metadata.name}')

echo {JENKINS_OBSERVER_POD}
evtmanager-topology-jenkins-observer-84c456f58c-jxlrj
```

To copy the plug-in from the image:

```
oc cp JENKINS_OBSERVER_POD:/opt/ibm/asm-observer-plugin-2.0.0-SNAPSHOT.hpi asm-observer-
plugin-2.0.0-SNAPSHOT.hpi

ls asm-observer-plugin-2.0.0-SNAPSHOT.hpi
asm-observer-plugin-2.0.0-SNAPSHOT.hpi
```

Obtain credentials

The credentials for the API are dynamically created. If IBM Cloud Pak for AIOps is installed with a release name of **asm**, then:

```
CREDS_USER=(oc get secret evtmanager-topology-asm-credentials -o jsonpath={.data.username} |
base64 -d)
CREDS_PASS=(oc get secret evtmanager-topology-asm-credentials -o jsonpath={.data.password} |
base64 -d)
```

Use the following command to print the UI API credentials:

```
echo CREDS_USER CREDS_PASS
```

Example user and password credentials returned:

```
aimanager-topology-aiops-user N8gJJGEfmmDnF6Q/1zg8NyAGKgQ9PmZQLhUSKd9/j54=
```

Obtain certificate

The following sample steps apply to the default router deployed on Red Hat OpenShift Container Platform. If more routers are used, adjust these steps. You perform these steps as **cluster administrator**.

Obtain the certificate:

```
oc get secret -n openshift-ingress router-certs-default -o jsonpath='{.data.tls\.crt}' |
base64 -d > ocp-default-router.crt
```

Obtain the certificate key:

```
oc get secret -n openshift-ingress router-certs-default -o jsonpath='{.data.tls\.key}' |
base64 -d > ocp-default-router.key
```

Convert to PKCS12:

```
openssl pkcs12 -export -out ocp-default-router.p12 -inkey ocp-default-router.key -in ocp-
default-router.crt
Enter Export Password: enter-suitably-secure-password
Verifying - Enter Export Password: enter-suitably-secure-password
```

Ensure that the files you have extracted from Red Hat OpenShift Container Platform are securely deleted.

Obtain observer URL

Obtain the hostname as in the following example:

```
oc get routes -l app.kubernetes.io/name=jenkins-observer -o jsonpath='{.items[*].spec.host}'
```

The Jenkins plug-in defines Jenkins pipeline DSL steps that you use to instruct the Jenkins builds to generate and then send topology data to the Jenkins Observer. Use the DSL steps to send the notification to include Artifactory BuildInfo modules information, in order for the build products to be modeled. You create an observer instance, which then triggers the sending of the notification to the observer.

For information on configuring Jenkins Observer jobs, see the related tasks.

Important: You perform the following steps as **Jenkins administrator**.

Install the Jenkins plug-in

1. Log in to your Jenkins server with a user ID that has administration (management) permissions.
2. Navigate to **Manage Jenkins** > **Manage Plugins** and select the **Advanced** tab.
3. In the Upload Plugin section, upload and install the provided plug-in (asm-observer-plugin-2.0.0-SNAPSHOT.hpi)

Configure the Jenkins plug-in

1. Navigate to **Manage Jenkins** > **Configure System** and select the **Agile Service Manager plugin**.
2. Enter the Jenkins Observer URL obtained earlier in the **Observer URL** field.

3. Create credentials for IBM Cloud Pak for AIOps using the **Manage Jenkins** > **Credentials** configuration page.

   Enter the credentials and the certificate details obtained earlier from the (or as the) cluster administrator.

   Auth Credentials Enter the username and password to authenticate with the Jenkins Observer at API level.

   Click **ADD**.

   Certificate Enter the PKCS12 format certificate to secure the communication between the Jenkins plug-in and the observer using SSL.

   Click **ADD**.

   For more detailed information on using the Jenkins Credentials feature, see the Jenkins site: https://jenkins.io/doc/book/using/using-credentials/

4. Enter a **Default Tenant ID**.

   The tenant to be used as default when sending information to IBM Cloud Pak for AIOps (if it is not overridden in the build).

5. Enter a **Default Job ID**.

   The default job ID to be used when sending information to IBM Cloud Pak for AIOps (if it is not overridden in the build). This represents the ID of the job the observer is running to process your information.

6. Click **TEST CONNECTION** to attempt to establish a connection with the observer using the configured credentials.

   If successful, a Connection successful! message is displayed. If unsuccessful, an error is returned.

   Note: The connection test does not verify the default job and tenant IDs.

Configuring your Jenkins build to notify IBM Cloud Pak for AIOps

After configuring the installed plug-in in Jenkins, you integrate the provided functionality into your existing Jenkins build. The following two steps describe how to use scripted pipeline syntax to send your build information to the observer.

Remember: You perform the following two steps on your Jenkins build (and **not** the Jenkins Observer).

1. Create an instance of the IBM Cloud Pak for AIOps observer communication object using the default credentials set during the configuration steps.

   ```
   def <my observer> = ASM.newObserver()
   ```

   Tip: Override credentials: to override any configured credentials (that is, the Jenkins credentials ID rather than the actual credentials themselves), you can use the following syntax:

   ```
   def <my observer> = ASM.newObserver
   authCredsId: 'auth-credentials-id', certCredsId: 'cert-credentials-id'
   ```

2. Send a build notification using the following syntax:

   ```
   ASM.notifyASM asmObserver: <my observer>
   ```

   Tip: You can override the default values for the tenant and job IDs:

   ```
   ASM.notifyASM tenantId: '<tenantId>', jobId: '<jobId>', asmObserver: <my observer>
   ```

   You can integrate the Jenkins plug-in with the Artifactory component using the modules information to model any artifacts the build has published to the repository:

   ```
   ASM.notifyASM asmObserver: <my observer>, artModules: <buildInfoObject>.getModules()
   ```

The following Jenkins file code sample shows the integration with IBM Cloud Pak for AIOps. It starts with the declaration of the observer instance, and ends with the notification to IBM Cloud Pak for AIOps. This notification should be added as a **`post / always`** block, so that the information is sent as the last operation of the build, and is sent regardless of the result of the build.

The optional **try / catch** around the notification to IBM Cloud Pak for AIOps is included in this sample to avoid the build failing if the notification to IBM Cloud Pak for AIOps fails.

The following sample also depicts the use of Artifactory to publish the build products, and how that information can be included in the notification to IBM Cloud Pak for AIOps using the artModules parameter.

```
**def asm = ASM.newObserver()**
def buildInfo = Artifactory.newBuildInfo()
buildInfo.env.capture = true

… your build logic here …
**
post {
    always {
      script {
            try {
                 ASM.notifyASM asmObserver: asm, artModules: buildInfo.getModules()
            } catch (err) {
                echo An error ocurred sending a notification to ASM:  {err}"
            }
          }
      }
   }
}**
```

Tip: You can further refine your integration and visualization.

After you install and configure the Jenkins plug-in, configure the Jenkins Observer listen job.

You can further refine your IBM Cloud Pak for AIOps system to take advantage of its topology and other customization functionality. For more information, see Customizing the Jenkins topologies.

You can extend your Jenkins integration with rules to merge data from different sources, custom topology display conventions, and the use of templates for the automated generation of topologies. For more information, see Additional Jenkins plug-in customization tasks.

More Jenkins plug-in customization tasks

You can extend your Jenkins integration with rules to merge data from different sources, custom topology display conventions, and the use of templates for the automated generation of topologies.

You first install and configure the Jenkins plug-in, before extending its functionality as described here.

Trouble: For useful Jenkins troubleshooting information, see troubleshooting.

Customizing the Jenkins topologies

You can customize how your Jenkins topologies are displayed in a number of ways, either by using the Jenkins customizations supplied with IBM Cloud Pak® for AIOps, or by creating your own custom display elements.

Creating Jenkins topology templates

Over time topologies generated through a specific Jenkins Observer job may grow, including the information from different Jenkins builds. You can use the IBM Cloud Pak for AIOps topology template functions to create meaningful, scoped views per build.

Example

The information obtained from Jenkins through the Red Hat OpenShift Container Platform plug-in for Jenkins and the Jenkins Observer includes URLs for several Jenkins elements. You can create topology tools that use these URLs to provide in-context

navigation to Jenkins or your Git repositories from within the topologies displayed in Red Hat OpenShift Container Platform. The code snippets in this section demonstrate how to create such tools.

If required, you can customize how your Jenkins topologies are displayed further using the Configuring topology elements topics.

Opens a GitHub repository commits log

```
function url2Http(url) {
  if(url.indexOf("http") != -1) {
      return url;
  }
  var res = url.split(":");
  var ghUrl = res[0].replace("git@", "https://");
  var repoUrl = res[1].replace(".git", "");
  return ghUrl + "/" + repoUrl;
}

if (asmProperties.repositoryUrl && asmProperties.revision) {
    var normUrl = url2Http(asmProperties.repositoryUrl);
    window.open(normUrl + '/ commits/'
        + asmProperties.revision);
}
```

Relationship type is **builds**

Opens the specific commit

```
function url2Http(url) {
  if(url.indexOf("http") != -1) {
      return url;
  }
  var res = url.split(":");
  var ghUrl = res[0].replace("git@", "https://");
  var repoUrl = res[1].replace(".git", "");
  return ghUrl + "/" + repoUrl;
}

if (asmProperties.repositoryUrl && asmProperties.revision) {
    var normUrl = url2Http(asmProperties.repositoryUrl);
    window.open(normUrl + '/commit/'
        + asmProperties.revision);
}
```

Relationship type is **builds**

Opens the Jenkins resources records

```
if (asmProperties.url) {
    window.open(asmProperties.url);
}
```

Resource types are **person**, **build**, **job**

Opens a GitHub repository tree view from the commit

```
function url2Http(url) {
  if(url.indexOf("http") != -1) {
      return url;
  }
  var res = url.split(":");
  var ghUrl = res[0].replace("git@", "https://");
  var repoUrl = res[1].replace(".git", "");
```

```
  return ghUrl + "/" + repoUrl;
}

if (asmProperties.repositoryUrl && asmProperties.revision) {
    var normUrl = url2Http(asmProperties.repositoryUrl);
    window.open(normUrl + '/tree/'
        + asmProperties.revision);
}
```

Relationship type is **builds**

Opens the build log console view in Jenkins

```
if (asmProperties.buildConsoleLog) {
    window.open(asmProperties.buildConsoleLog);
}
```

Resource type is **build**

Creating Jenkins topology templates

Over time topologies generated by a specific Jenkins Observer job may grow, including the information from different Jenkins builds. You can use the IBM Cloud Pak® for AIOps group templates functionality to create meaningful, scoped groups per build. This can then be included as part of a broader application definition.

Create 'per build' topology templates

The following template generates a group for each build, which includes the build and its associated pipeline resources:

1. Log in to the IBM Cloud Pak Automation console.
2. From the main navigation, click **Application management**.
3. Click the **Group templates** icon on the upper right corner of the page.
4. From the **Group templates** page, click **Create a new template**.
5. Select **Dynamic template** to ensure that all matching builds will generate topologies like this.
6. Click **Start** to begin building your template.
7. On the Preview pane, search for the build resource and select **Follow Relationship**. Repeat this for all of the connected resources in your build topology.

   Important: Do not expand the 'lastRunOf' relationship.
8. Set the Group type as 'cicd'.
9. (Optional) Define tags for the template, then click **Save template & generate groups**.

Create 'current pipeline status' group templates

The following template generates a group that shows the status of your pipeline that is the latest build for which a notification was received by the observer and associated resources:

1. Log in to the IBM Cloud Pak Automation console.
2. From the main navigation, click **Application management**.

3. Click the **Group templates** icon  on the upper right corner of the page.
4. From the **Group templates** page, click **Create a new template**.
5. Select **Dynamic template** to ensure that all matching builds will generate topologies like this.
6. Click **Start** to begin building your template.
7. On the Preview pane, search for the job resource and select **Follow Relationship**. Then click 'lastRunOf'. The latest build resource for which a topology was generated is displayed. Expand any relationships you want to include in this template.

Important: Do not expand the 'RunOf' relationship.

6. Set the Group type as 'cicd'.
7. (Optional) Define tags for the template, then click **Save template & generate groups**.

Configuring Juniper CSO Observer jobs

Using the Juniper CSO Observer, you can define a full load job that will gather and read data about topology data from Juniper CSO.

Before you begin

Important: The Juniper CSO observer supports the on-premises Juniper CSO version 4.1.0.

Ensure you have the Juniper CSO details to hand, such as details of the Juniper CSO apiserver and its credentials.

The Juniper CSO Observer retrieves topology data from Juniper CSO Release 4.1 through REST APIs exposed by CSO API server.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

You define and start the following job.

Full Topology Upload job

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) manually from the Observer UI, or schedule them to run at set times when configuring them.

Run this job whenever you need the Juniper CSO topology data refreshed.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required

Parameter	Action	Details
Juniper CSO Central MS URL	Specify the Juniper CSO Central micro-service URL	Required
CSO Keystone Authentication URL	Enter the authentication URL for the identity service	Required
CSO user domain name	Enter the CSO user domain name	Required
CSO domain or project or tenant name	Enter the CSO domain or project or tenant name	Required
CSO Authentication username	Specify the CSO authentication username	Required
CSO Authentication password	Specify the CSO authentication password	Required
Enable/Disable Secure Connection to CSO Host	Set to 'true' to secure the connection, otherwise set to 'false' to bypass	Required
Target system certificate	Specify the certificate file name.	Required if **`Enable/Disable Secure Connection to CSO Host`** is set to 'true'. To obtain authentication certificates using OpenSSL and store them as secrets, see Configuring observer jobs security.
SSL truststore File	If **`Enable/Disable Secure Connection to CSO Host`** is set to 'true', then supply the HTTPS truststore file name	Optional
SSL truststore Password	If **`Enable/Disable Secure Connection to CSO Host`** is set to 'true', then supply a password to decrypt the HTTPS truststore file	Optional
SSL Validation	Choose whether SSL validation is on or off. Turning SSL validation off will use HTTPS without host verification.	Optional
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Configuring Kubernetes Observer jobs

Using this observer, you can configure jobs that discover the structure of your Kubernetes clusters, including pods, worker nodes and containers.

Before you begin

Important: The Kubernetes Observer supports Kubernetes version 1.25.8.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The observer reads topology data from Kubernetes through its REST APIs, or Weave Scope.

You can run the following jobs:

Load

A transient (one-off) job that loads all requested topology data from a Kubernetes environment.

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) manually from the Observer UI, or schedule them to run at set times when configuring them.

Tip: When observing different Kubernetes clusters, make sure that the name of the data center is unique across all of the jobs.

Local

Note: The Kubernetes Observer Local job has been deprecated as of IBM Cloud Pak for AIOps 3.4.0. Use the Load job instead to performs local observations of the Kubernetes REST API for available resources.

Weave_scope Listen

A stand-alone job that listens to the Weave Scope agent and continues to stream topology and state data to IBM Cloud Pak® for AIOps.

The Weave Scope listen job provides visibility of your Kubernetes services, pods, containers, deployments, stateful sets, Cron Jobs and processes for a specified namespace.

A long-running job that monitors its source for updates and runs until it is explicitly stopped, or until the Observer is stopped.

Kubeconfig Load

A stand-alone load job that accepts kubeconfig data in JSON format to performs a one-off discovery.

Kubernetes Observer Load job prerequisites

Before defining a Kubernetes Observer Load job, you must create a service account in the Kubernetes environment and obtain its token. The following example uses the default namespace.

1. Create either cluster role access or role access.

   For clusterole access, create a configuration file called **`asm-k8s-observer.yaml`** with the custom cluster role **`asm:kubernetes-observer`**.

   Use the following sample content:

```
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  annotations:
    rbac.authorization.kubernetes.io/autoupdate: "true"
  name: asm:kubernetes-observer
rules:
- apiGroups: [""]
  resources: [ "pods", "namespaces", "nodes", "services", "endpoints",
"persistentvolumes", "persistentvolumeclaims"]
  verbs: ["get", "list"]
- apiGroups: ["apps"]
  resources: ["replicasets", "deployments", "statefulsets", "daemonsets"]
  verbs: ["get", "list"]
```

For role access:

Use the following sample content:

```
kind: Role
apiVersion: rbac.authorization.k8s.io/v1
metadata:
  namespace: default
  name: asm:kubernetes-observer
rules:
- apiGroups: [""]
  resources: [ "pods", "services", "endpoints"]
  verbs: ["get", "list"]
- apiGroups: ["apps"]
  resources: ["replicasets", "deployments", "statefulsets", "daemonsets"]
  verbs: ["get", "list"]
```

Observing Custom Resources:

To model custom resource definitions (CRDs) and custom resources (CRs) in the topology, identify the names of the CRDs that are present in the environment by running the **`oc get crds`** command. Then, update the job parameters to include the names of CRDs to be discovered. When a CRD is being included in the job, the respective CRs will also be discovered.

Example (discovering ibm.com CRDs):

```
oc get crds | grep ibm.com
```

Sample output:

```
asmformations.asm.ibm.com                                         2020-12-03T01:53:38Z
asms.asm.ibm.com                                                  2020-12-03T01:53:38Z
cemformations.cem.ibm.com                                         2020-12-03T01:53:38Z
cemserviceinstances.cem.ibm.com                                   2020-12-03T01:53:38Z
noiconnectionlayers.noi.ibm.com                                   2020-12-03T01:47:19Z
noiformations.noi.ibm.com                                         2020-12-03T01:47:19Z
noihybrids.noi.ibm.com                                            2020-12-03T01:47:19Z
nois.noi.ibm.com                                                  2020-12-03T01:47:20Z
```

Also, change the existing **`ClusterRole`** in the target environment, to include the following resources:

```
- apiGroups: ["apiextensions.k8s.io"]
  resources: ["customresourcedefinitions"]
  verbs: ["get", "list"]
- apiGroups: ["noi.ibm.com"]
  resources: ["nois", "noihybrids", "noiformations", "noiconnectionlayers"]
  verbs: ["get", "list"]
- apiGroups: ["asm.ibm.com"]
  resources: ["asms", "asmformations"]
  verbs: ["get", "list"]
- apiGroups: ["cem.ibm.com"]
  resources: ["cemformations"]
  verbs: ["get", "list"]
```

The **`apiextensions.k8s.io`** segment of the cluster role provides your permission required to access all the custom resource definitions resources.

```
- apiGroups: ["apiextensions.k8s.io"]
  resources: ["customresourcedefinitions"]
  verbs: ["get", "list"]
```

The **`asm.ibm.com`** segment provides access to **`asms`** and **`asmformations`** custom resources. The names of the resources listed are the plural for each of the resources. They are usually derived from the first word of the CRD.

```
- apiGroups: ["asm.ibm.com"]
  resources: ["asms", "asmformations"]
  verbs: ["get", "list"]
```

For a fresh discovery:

Create a configuration file called **asm-k8s-observer.yaml** in the target environment using the yaml resource provided in the sample content, then run the following command before moving to the next step:

```
oc create -f asm-k8s-observer.yaml
```

For an existing discovery:

Create a configuration file called asm-k8s-observer.yaml using the yaml resource provided in the sample, then run the following command to patch the clusterole:

```
oc patch clusterrole asm:kubernetes-observer --patch "$(cat asm-k8s-observer.yaml )"
```

2. Create a service account and then assign the service account to the same namespace as kubectl:

   ```
   oc create serviceaccount asm-k8s-account
   ```

   To verify that the service account exists:

   ```
   oc get serviceaccount
   ```

3. Bind the **asm:kubernetes-observer** ClusterRole or role to the **asm-k8s-account** service account.

   The command binds the ClusterRole or role to the service account that is created on the 'default' namespace.

   Example command for clusterole binding:

   ```
   oc create clusterrolebinding asm-k8s --clusterrole=asm:kubernetes-observer --
   serviceaccount=default:asm-k8s-account
   ```

   Example command for role binding:

   ```
   oc create rolebinding asm-k8s --role=asm:kubernetes-observer --
   serviceaccount=default:asm-k8s-account
   ```

   Output:

   ```
   rolebinding.rbac.authorization.k8s.io/asm-k8s created
   ```

4. Obtain the Kubernetes service account token for observation by completing the following steps:

   1. Look for the mounted service account token secret from the secrets section:

      ```
      oc describe serviceaccount asm-k8s-account
      ```

      Example:

      ```
      oc describe serviceaccount asm-k8s-account
      Name: asm-k8s-account
      Namespace: default
      Labels: <none>
      Annotations: <none>
      Image pull secrets: asm-k8s-account-dockercfg-wwzx2
      Mountable secrets: asm-k8s-account-dockercfg-wwzx2
      Tokens: asm-k8s-account-token-tzl52
      Events: <none>
      ```

   2. Describe the asm-k8s-account-token-******* (which in this example is tzl52) to obtain the token's value:

      ```
      oc describe secret asm-k8s-account-token-tzl52
      ```

   For Kubernetes load jobs, ensure you have the Kubernetes service details to hand, such as the Kubernetes host IP and SSL Certificate details.

Kubernetes Observer Weave Scope listen job prerequisites

For Weave Scope listen jobs, do a Weave Scope installation, and then configure a job using the Weave Scope IP and port parameters. You must install Weave Scope and then use the Weave Scope Master IP and Node port parameters. For more information on Weave Scope, see the following location: https://www.weave.works/docs/scope/latest/introducing/

1. Create namespace 'weave' with 'ibm-privileged-psp'.

```
oc create namespace weave
oc -n weave create rolebinding weave-clusterrole-rolebinding --clusterrole=ibm-
privileged-clusterrole --group=system:serviceaccounts:weave
```

2. Install Weave Scope using the following command:

   ```
   oc apply -f "https://cloud.weave.works/k8s/scope.yaml?k8s-service-type=NodePort&k8s-
   version=$(kubectl version | base64 | tr -d '\n')"
   ```

   This will result in a port being opened that the observer can use.

3. You can discover the NodePort using the following command:

   ```
   oc -n weave describe service weave-scope-app
   ```

4. Launch the Weave Scope UI using the following URL:

   ```
   https://<master ip>:<NodePort>
   ```

5. Create a new route using the Red Hat OpenShift Container Platform console to expose the Red Hat OpenShift Container Platform Weave Scope service under the weave namespace. In the left panel of the Red Hat OpenShift Container Platform console, click **Networking** > **Routes**.

6. Click **Create Route** to create a new route.

7. Complete the required fields and then click **Create**.

8. A new route has been created. Launch the Weave Scope UI using the location link.

Kubernetes Observer kubeconfig load job prerequisites

For a kubeconfig load job, ensure your kubeconfig.yml file is in the correct format. If it is not, generate a new file.

Important: The kubeconfig should meet the following criteria:

- Single cluster
- User type of 'token'
- Base64 encoded JSON

View the kubeconfig.yml file:

```
oc config view --raw -o json
```

Important: Make sure the token and certificate are not omitted or redacted.

If it meets all required conditions, encode it using the following command:

```
oc config view --raw -o json | base64 -w 0
```

If it does not meet all conditions, generate a new kubeconfig.yml file using the following steps:

1. Create a cluster-role or role, service account, cluster-role/role binding. From the service account, get the secret, and from the secret get the token. Refer to "Kubernetes Observer Load job prerequisites" for detailed steps.

2. Get the current context details:

   ```
   oc config current-context
   ```

3. Get the current cluster, cluster ca (certificate-authority-data), and cluster server details:

   ```
   oc config view --raw -o json
   ```

4. Create a kubeconfig.yml using the obtained data. Sample kubeconfig.yml file:

   ```
   apiVersion: v1
   kind: Config
   current-context: <CURRENT_CONTEXT>
   contexts:
   - name: <CURRENT_CONTEXT>
     context:
   ```

```
    cluster: <CURRENT_CONTEXT>
    user: <USER>
    namespace: <NAMESPACE>
clusters:
- name: <CURRENT_CONTEXT>
  cluster:
    certificate-authority-data: <CLUSTER_CA>
    server: <CLUSTER_SERVER>
users:
- name: <USER>
  user:
    token: <TOKEN_VALUE>
```

Where **USER** is the user-defined value, and **NAMESPACE** is the namespace of the service account.

5. Test the kubeconfig validity and access:

```
oc --kubeconfig <config>.yml get all
```

6. Run the following command to convert the file to base64 encoded json:

```
oc --kubeconfig <config>.yml config view --raw -o json | base64 -w 0
```

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Load job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job. **Note:** The connection name cannot be the same as any of the following observer job names: 'local', 'weave_scope', 'load', 'kubeconfig'.	Required
data_center	Specify the name of the data center in which the Kubernetes instance is running. The data center name is treated as the provider of the data. **Note:** If you change the data center name, the observation is treated as a separate observation under the different provider, and a duplicate set of resources and groups will be created under the new provider.	Required
Kubernetes token	The service account token for Kubernetes	Required
Role permission token	Determine whether the provided token is a role token. Default to false for cluster role permissions.	Optional
Kubernetes Master IP address	Enter the Kubernetes Master IP address	Required. Get the Kubernetes master IP and its API port using the 'oc cluster-info' command.
Kubernetes API port	Enter the Kubernetes API port number	Required
Trust all certificate by bypassing certificate verification	Enter **true** if you want to connect to Kubernetes without certificate	Required

Parameter	Action	Details
Exact HTTPS certificate file name	Enter the exact name of the SSL/HTTPS certificate	Optional. If 'Trust all certificate' is set to **false**, then this parameter is Required.
SSl Validation	Choose whether SSL validation is **true** or **false**. If **false**, HTTPS without hostname validation is used.	Optional
API query timeout (ms)	Specify the Kubernetes REST API query timeout	Optional. The default is 5000 ms (that is, 5 seconds)
Correlate	If 'true', enables the Event Analytics correlation on the namespace groups	Optional
Names of the custom resource definitions	A number of custom resource definitions (CRDs) that can be queried from the Kubernetes API	Optional
Namespace	Specify the Kubernetes namespace	Optional. Specify the Kubernetes namespace to observe, leave blank or enter `*` to observe all.
Terminated pods	Choose whether terminated pods should be hidden (**true** or **false**)	Optional. The default is false.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Set the time when the job should run, and whether it should run at regular intervals. By default, the job runs immediately, and only once. Optionally, you can specify a future date and time for the job to run, and then set it to run at regular intervals after that, if required. The run intervals must be at least 90 seconds apart, and if you set them at less than 15 minutes, a warning is displayed, as the frequency can impact system performance.	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Weave_scope job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job. **Note:** The connection name cannot be the same as any of the following observer job names: 'local', 'weave_scope', 'load', 'kubeconfig'.	Required
Host	Enter the Weave Scope hostname (or IP address) of the web socket to be observed	Required
Port	Enter the Weave Scope port number of the web socket to be observed	Required. The Port can be obtained using the 'oc -n weave describe service weave-scope-app' command.
Cluster Name	Enter the name of the cluster or data center to be observed	Required

Parameter	Action	Details
Namespaces	Enter a list of namespaces to be observed	Optional. If left empty, all namespaces will be observed.
Resource types	Select the Weave Scope resource types to observe	Optional.
Resources to exclude	List resources to be excluded by ID, label, rank or namespace	Optional. Containers named 'pod' are excluded by default.
Refresh interval	Specify the refresh interval in seconds between updates. The minimum allowed value is 60s, and the default value is 120s.	Optional
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional

Kubeconfig load job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job. **Note:** The connection name cannot be the same as any of the following observer job names: 'local', 'weave_scope', 'load', 'kubeconfig'.	Required
data_center	Specify the name of the data center in which the Kubernetes instance is running. The data center name is treated as the provider of the data. **Note:** If you change the data center name, the observation is treated as a separate observation under the different provider, and a duplicate set of resources and groups will be created under the new provider.	Required
kubeconfig	Provide the contents of the kubeconfig in json format. The kubeconfig needs to be a json with a single cluster and its user needs to be of token type.	Required. The json needs to be base64 encoded.
Terminated pods	Choose whether terminated pods should be hidden (**true** or **false**)	Optional. The default is false.
API query timeout (ms)	Specify the Kubernetes REST API query timeout	Optional. The default is 5000 ms (that is, 5 seconds)
Correlate	If 'true', enables the Event Analytics correlation on the namespace groups	Optional
Names of the custom resource definitions	A number of custom resource definitions (CRDs) that can be queried from the Kubernetes API	Optional
Role permission token	Determine whether the provided token is a role token. Default to false for cluster role permissions.	Optional
Namespace	Specify the Kubernetes namespace	Optional. Specify the Kubernetes namespace to observe, or `*` to observe all. Leaving this field blank will observe the namespace in which IBM Cloud Pak for AIOps is installed. When using a role permission token, specify the single namespace where the role was created.

Parameter	Action	Details
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Set the time when the job should run, and whether it should run at regular intervals. By default, the job runs immediately, and only once. Optionally, you can specify a future date and time for the job to run, and then set it to run at regular intervals after that, if required. The run intervals must be at least 90 seconds apart, and if you set them at less than 15 minutes, a warning is displayed, as the frequency can impact system performance.	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Configuring Network Manager Observer jobs

Using the IBM Tivoli Network Manager Observer, you can define jobs that dynamically load data discovered by Network Manager for analysis.

Before you begin

Important: The IBM Tivoli Network Manager Observer supports the on-premises IBM Tivoli Network Manager version 4.2.

Ensure you have the IBM Tivoli Network Manager service details to hand, such as the domain, host and port number.

The IBM Tivoli Network Manager Observer jobs extract Network Manager resources using an Object Query Language JDBC driver, and the observer loads and updates the resources and their relationships within the topology.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

You configure the following two jobs.

Load

A transient (one-off) job that queries Network Manager for topology data, and performs a complete upload for a single IBM Tivoli Network Manager domain.

By default, Load jobs are one-off, transient jobs that upload all of the requested topology data as soon as they are triggered.

For cross-domain connections: Each ITNM domain must read existing cross-domain connections from the topology service, which means that you may need to run two observations of all domains contributing cross-domain connections (that is, run the full load job twice) before all cross-domain edges are created.

You can also run these jobs (again) manually from the Observer UI, or schedule them to run at set times when configuring them.

Listen

A long-running job that monitors the Network Manager message bus for changes and updates the topology service. When the job is started, the observer creates an OQL connection that listens for changes in the IBM Tivoli Network Manager network. Any resources added, changed or deleted are passed on by the OQL connection and the topology is updated.

The listen job runs until it is explicitly stopped, or until the observer is stopped.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
ITNM instance name	Specify the IBM Tivoli Network Manager instance name	Required. Specify the same instance name for all jobs to enable connectivity across domains.
ITNM domain	Specify the IBM Tivoli Network Manager domain	Required
Hostname or Server IP	Specify the IBM Tivoli Network Manager hostname or server IP on which the domain is running	Required
ITNM domain port	Specify the IBM Tivoli Network Manager port for the specified domain	Required. The value of `port` will vary if multiple domains exist. To determine which port number to use for a Network Manager domain, look for the domain-specific ncp_config entry in the `$NCHOME/etc/precision/ServiceData.cfg` file.
OQL connection timeout (ms)	Specify the OQL Connection timeout value	Optional. The default is 3000 (30 seconds).
Exclude resources without connections	Select whether to display disconnected resources	Optional. The choices are **true** and **false**. The default is **true**.
Edge Type Map	Map IBM Tivoli Network Manager layers to topology relationship types	Optional. If blank, IBM Cloud Pak® for AIOps will auto-map. The value of topology type can be found in IBM Tivoli Network Manager in '$NCHOME/precision/disco/stitchers/DNCIM/PopulateDNCIMTopologies.stch'; alternatively, run the following OQL statement against the model service to list the available topology type: `select ENTITYNAME from ncimCache.entityData where METACLASS='Topology'`
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Configuring a New Relic Observer job

Use the New Relic Observer when you have a New Relic account with a New Relic Infrastructure subscription. Using this observer you can configure jobs that dynamically load New Relic Infrastructure resource data through New Relic for analysis.

Before you begin

Important: The New Relic Observer supports the cloud/SaaS New Relic version.

Ensure you have the New Relic account and New Relic Infrastructure subscription details to hand, such as the account name, account ID, and New Relic Insights API query key.

Restriction: New Relic applies a 1000 results limit on all New Relic Query Language (NRQL) queries. To accommodate this limit when retrieving data from the SystemSample, StorageSample, ProcessSample and NetworkSample event tables, the New Relic Observer uses the following NRQL query time clause:

```
"SINCE 4 hours ago LIMIT 1000"
```

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The Observer uses the New Relic Infrastructure subscription and makes active New Relic Query Language (NRQL) calls over REST to New Relic Insights to download New Relic Infrastructure resource data.

The New Relic Observer loads the following New Relic Infrastructure resources and their relationships:

- Host
- Storage
- OS
- Network Interfaces
- Processes

The New Relic Observer job extracts New Relic Infrastructure resources from New Relic using New Relic Query Language (NRQL) over REST, and the observer loads and updates the resources and their relationships.

You configure the following job.

Load: By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: For more information, see Configuring observer job security

Table. Parameter details

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
New Relic Name	Specify the New Relic account name or tenant name	Required
New Relic account ID	Specify the New Relic account ID	Required.
New Relic Insights Query API key	Specify the New Relic Insights Query API key	Required.

Parameter	Action	Details
filterCriteria	Extend the result set returned to IBM Cloud Pak® for AIOps	Optional. The default value is `SINCE 4 hours ago LIMIT 1000`. For more information, see the documentation for New Relic Query Language.
Generate debug support file	Set this parameter to `True` in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to `False` after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Tips:

- **New Relic account ID**: To obtain the account ID log in to New Relic and then go to the URL https://rpm.newrelic.com/accounts/%3CaccountId%3E
- **New Relic Insights Query API Key**: A New Relic user with a New Relic Infrastructure subscription is required to generate a New Relic Insights Qeury API Key. For more information, see https://docs.newrelic.com/docs/insights/insights-api/get-data/query-insights-event-data-api.

Configuring OpenStack Observer jobs

Using the OpenStack Observer, you can configure jobs that dynamically load OpenStack data for analysis.

Before you begin

Important: The OpenStack Observer supports the on-premises OpenStack up to versions Ocata, Pike, Rocky, Queens, Stein, Train, Zed and Antelope.

Ensure you have the OpenStack service details to hand, such as the parameters for its APIs or RabbitMQ message bus. If you are configuring a query job, have OpenStack location and authorization details to hand. If you are configuring a rabbitmq job, you must also identify and provide access to the RabbitMQ message bus.

OpenStack installation requirements

If you previously installed OpenStack with DevStack, you must add the configuration that is specified here to the end of the local.conf file, and reinstall it. If you previously installed OpenStack with another installation method, you must add the configuration that is specified here to the nova.conf file, and then restart the Nova (compute) service.

- If you **have already** installed OpenStack using DevStack, add the following configuration to the end of the local.conf file, and **then reinstall** OpenStack.
- If you are **planning to** install OpenStack using DevStack, add the following configuration to the end of the local.conf file **before installation**.

```
[[post-config|$NOVA_CONF]]
[DEFAULT]
notification_topics = notifications,com.ibm.asm.obs.nova.notify
notification_driver=messagingv2
notify_on_state_change=vm_and_task_state
notify_on_any_change=True
```

For standard (or any other) OpenStack installations, add the following configuration under the `[DEFAULT]` section of the nova.conf file, and then you restart the Nova (compute) service.

```
notification_topics = notifications,com.ibm.asm.obs.nova.notify
notification_driver=messagingv2
notify_on_state_change=vm_and_task_state
notify_on_any_change=True
```

Note: OpenStack uses RBAC-based protection of its API by defining policy rules based on an RBAC approach. Availability of resources retrieved by the observer is also governed by this same policy. For example, a VM created in project A by users with the admin role may be available to other users with the same admin role. This can be configured or modified according to user requirements in the OpenStack's policy configuration.

OpenStack port numbers

OpenStack assigns standard port numbers to each of its service. To communicate with the OpenStack instance, you must enable the following default ports.

Tip: Check with your OpenStack administrator if custom port numbers have been configured.

Load jobs default ports:

```
- KeyStone: 5000
- Nova: 8774
- Neutron: 9696
- Glance: 9292
- Cinder: 8776
- Heat: 8004
```

Listen jobs default port:

```
- RabbitMQ: 5672
```

Troubleshooting

A **Certificate Chaining Error** can occur when launching an OpenStack Observer job.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The OpenStack Observer jobs extract OpenStack resources through REST or RabbitMQ, and the observer loads and updates the resources and their relationships within the topology service.

You configure and run the following two jobs.

Restapi Load job

A transient (one-off) job that loads all requested topology data from the OpenStack instance by REST API.

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

The job loads baseline topology data through the following OpenStack's APIs:

- Keystone (identity)
- Cinder (block storage)
- Glance (image)
- Heat (orchestration)
- Neutron (network)
- Nova (compute)

Restriction: An OpenStack environment that has a list of endpoints whereby the 'heat-cfn' service comes first (before the 'heat' service) will encounter a JSON parsing error recorded in the logs due to a known issue in the openstack4j library. When this happens, the full load for the heat service will be skipped entirely. Other services will run as normal.

Rabbitmq Listen job A long-running job that reads messages on OpenStack's RabbitMQ message bus for activity from the Cinder (block storage), Heat (orchestration), Neutron (network) and Nova (compute) components continually, until it is explicitly stopped, or until the Observer is stopped.

Note: The rabbitmq job should only be run after an initial restapi job has been completed.

Restriction: Only one rabbitmq job should be listening to one queue (or sets of queues) at any one time. If you need to listen to multiple projects, then separate queues must be set up in OpenStack, with appropriate names, before separate listening jobs are submitted for each. For example, for Nova through the rmq_nova_notify attribute, for Neutron through the rmq_neutron_notify attribute.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Restapi Load job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
OpenStack authentication type	Specify the OpenStack connection authentication technique to use	Required. Choose either **v2_Tenant**, **v3_Unscoped**, **v3_Project**, **v3_Domain**, or **v3_ProjectDomain**.
OpenStack password	Specify the OpenStack password with which to authenticate	Required
OpenStack identity endpoint	Specify the authentication URL	Required. Must include the port and version.
Data center name	Specify the name of the data center in which the OpenStack instance is running	Required. If more than one OpenStack instance is run, and duplicate project or tenant names exist, you must disambiguate them here.
OpenStack username	Specify the OpenStack username to connect as (or to)	Required
OpenStack project name	Specify the OpenStack project name(s) to use, for example the value of **OS_PROJECT_NAME**. Multi project names should be separated by comma.	Required
OpenStack project domain name	Specify the project domain name to use, for example the value of **OS_PROJECT_DOMAIN_NAME**.	Optional
OpenStack user domain name	Specify the user domain name to use, for example the value of **OS_USER_DOMAIN_NAME**.	Optional
OpenStack region name	Specify the OpenStack region to use, for example the value of **OS_REGION_NAME**. The value is case-sensitive. If it is not set, the default value is *regionOne*.	Required for v3 authentication.
OpenStack perspective	Select the URL perspective the API accesses data from	Optional. Choose from **Admin**, **Public**, and **Internal**.

Parameter	Action	Details
Connection and read timeout (ms)	Choose the timeout setting for the connection and read actions	Optional. The default is 5000 (5 seconds).
Optionally load hypervisors	Choose whether to turn the loading of hypervisors on or off. **Note:** To load hypervisors, admin privileges must be assigned to the user.	Optional. The default is **false** (off).
Optionally load Host Aggregates	Choose whether to turn the loading of Host Aggregates on or off. **Note:** To load host aggregates data, admin privileges must be assigned to the user.	Optional. The default is **false** (off).
SSL Verification	Choose whether to use SSL verification (true or false). If false, HTTPS is used, but **without** hostname validation.	Optional
OpenStack host certificate	Specify a certificate name to load into the truststore	Required
SSL truststore file name	Specify a truststore file name	Required
SSL truststore file password	Specify a truststore file password	Required
Proxy username	Specify the proxy username.	Optional
Proxy password	Specify the proxy password	Optional
Proxy host	Specify the proxy host to connect via	Optional
Proxy port	Specify the proxy port to connect via	Optional. Default is `8080`.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Rabbitmq Listen job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
RabbitMQ username	Specify the AMQP username to connect to the broker	Required
RabbitMQ password	Specify the password to use to connect to the broker	Required
RabbitMQ hosts	Enter a (comma-separated) list of hosts in the RabbitMQ cluster	Required. Add a comma-separated host, or list of hosts, (for example, `host1:5672,host2:5672`) to connect to in the RabbitMQ cluster. The first host connected to successfully is used. **Note:** Use IP addresses instead of hostnames.
Data center name	Specify the name of the data center in which the OpenStack instance is running	Required. If more than one OpenStack instance is run, and duplicate project or tenant names exist, you must disambiguate them here.
OpenStack username	Specify the OpenStack username to connect as (or to)	Required
OpenStack project name	Specify the OpenStack project	Optional

Parameter	Action	Details
RabbitMQ virtual host name	Specify the virtual host to connect to the broker	Optional
Use SSL?	Choose whether to use an SSL connection	Optional. Choose **true** or **false**. For RabbitMQ, you must choose **true**.
Nova v2 Oslo message queue	Specify the Nova v2 Oslo message queue	Optional
Neutron v2 Oslo message queue	Specify the Neutron v2 Oslo message queue	Optional
Cinder v2 Oslo message queue	Specify the Cinder v2 Oslo message queue	Optional
Heat v2 Oslo message queue	Specify the Heat v2 Oslo message queue	Optional
Number of consumer instances	Specify the number of consumer instances to create for each API queue type	Optional
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional

Important: You **must** specify the following properties consistently for both the restapi and rabbitmq jobs:

- Data center name
- OpenStack project name
- OpenStack username

Configuring Rancher Observer jobs

Using the Rancher Observer, you can define a full load job that will read data from Rancher through the REST API and generate a topology.

Before you begin

Important: The Rancher Observer supports the on-premises Rancher version 2.3.2 and API version 3.0.

Ensure you have the Rancher details to hand, such as the Rancher host IP, accessKey, secretKey and clusterId, before running the observer job.

The Rancher Observer supports a transient (one-off) Load job that loads all requested topology data through Rancher REST API to build the topology, and then exit.

You define and start the following job. You must edit the parameters in the configuration file before running this job.

Full Topology Upload job By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) manually from the Observer UI, or schedule them to run at set times when configuring them.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Rancher Host IP	Enter the Rancher host IP address	Required
Rancher Access Key ID	Supply the Rancher access key	Required
Rancher Secret Access Key	Supply the Rancher secret key	Required
Rancher Cluster ID	Enter the cluster ID to be discovered	Required
Enable/Disable Secure Host Connection to Rancher's Host	Set to 'true' to secure the connection. Alternatively, set to 'false' to bypass secure host connection.	Required
Certificate	If enable_secure_host_connection is set to 'true', update the global secret with encrypted certificate information. See the relevant step here.	Optional
SSL truststore File	If enable_secure_host_connection is set to 'true', provide the JKS name, and the observer creates the JKS file	Optional
Password SSL truststore File	If enable_secure_host_connection is set to 'true', set the password to decrypt the HTTPS truststore file.	Required if enable_secure_host_connection is set to 'true'. You **must** supply the SSL truststore Password to decrypt the HTTPS truststore file.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Configuring REST Observer jobs

Use the REST Observer for obtaining topology data through REST endpoints. This observer is a counterpart to the File Observer.

Before you begin

For the Swagger API documentation, see REST Observer for Agile Service Manager.

About this task

The REST Observer passes topology data using a RESTful set of interfaces, which provide REST APIs that enable the following functionality:

- Management of Listen and bulk-replace job types.
- The insert-update (HTTP POST) of resources.
- The insert-update (HTTP POST) of relationships.
- The insert-replacement (HTTP PUT) of resources.
- The deletion (HTTP DELETE) of resources.
- A REST API that supports the deletion (HTTP DELETE) of all relationships of a particular type from a specified resource.
- A REST API that supports the deletion (HTTP DELETE) of a specific relationship.

Restriction: Resources created through REST can have a provider, but not an observer.

Benefits: Using the REST Observer rather than the File Observer or Topology Service APIs includes the following benefits:

- The ability to provide data through HTTP REST Endpoints instead of files.
- The processing performed by all observers in their framework ensures that metadata about observations from observers is managed correctly.
- A simple way of deleting all edges of a given type on a resource or a specific edge instance.

To use the REST Observer, a job request must be issued (HTTP POST) to the Observer instance job management APIs before sending data to the Resource and Relationship APIs.

Listen

A long-running listen job capable of consuming topology data over a long period. A listen job is designed to support scenarios where the input data stream is unpredictable, or little or no consistency or versioning of resources exists within the data stream.

Synchronize (bulk replace)

A long-running job with the same resource replace semantics as the File Observer. Bulk-replace jobs are designed to support scenarios where a known set of resources are subject to updates or versioning, and a prior observation about resources is to be replaced with a new one. This job provides a new set of resources and relationships and synchronizes them to the topology service, causing any previous data provided by the observer to be deleted and replaced with the new data.

Note: In both cases, a provider job parameter is required to identify the origin of the data being provided to the observer job.

Once a job request has been successfully submitted, you can start to provide data to the Resource and Relationship APIs on behalf of a particular job instance.

The Resource and Relationship APIs may respond with an HTTP 503 Service Unavailable response with a Retry-After: 10 seconds in the header. This indicates that even though the request against those APIs is valid, the observer has not been able to ascertain that metadata about the job is held in IBM Cloud Pak® for AIOps yet; this may be due to, for example, any prevailing conditions in the network that support the micro-services.

Tip: If such a response is received, try the request again later.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Important: Ensure that the body is structured correctly. When posting the body, information included in the body after the closing **}** that matches an opening **{** is ignored, and no error is recorded.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Provider	Specify the name of the program or system to provide data	Required

Parameter	Action	Details
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. **Tip:** You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.

REST Observer API access

API name: topology-rest-observer

Public API URLs: **`/aiops/api/application-manager/topology-rest-observer/v1/*`**

For information on how to obtain route and token information, see Accessing the console.

For information on how to obtain your token, see Accessing APIs.

Swagger API documentation

For the Swagger API documentation, see Topology REST Observer API. The documentation explains how to use the REST Observer APIs.

Creating cross-data origin relationships

Using the REST Observer functionality, you can create edges between resources from different observations or providers.

It requires both the **`uniqueID`** and the provider of the resources at each end of the link (so typically applies to previously observed resources).

In the following syntax examples, note the **`_toProvider`** and **`_fromProvider`** edge properties.

You can define an edge from a resource in the current observation to one owned by a different provider. This example needs to be POSTed to the REST observer's **`/rest/resources`** endpoint:

```
{
   "uniqueId":"host1.ibm.com",
   "entityTypes":["host"],
   "name":"host1",
   "_references":
   [
       {
          "_edgeType":"connectedTo",
          "_toProvider":"someOtherProvider",
          "_toUniqueId":"host2.ibm.com"
       }
   ]
}
```

You can define an edge from a resource between two different providers. This example needs to be POSTed to the REST observer's **`/rest/references`** endpoint:

```
{
   "_fromProvider":"providerOne",
   "_fromUniqueId":"host1-interface1",
   "_edgeType":"connectedTo",
   "_toProvider":"providerTwo",
   "_toUniqueId":"host2-interface1"
}
```

Example listen job

The following examples show how to use the REST Observer listen job to create and adjust a small topology.

Start the Listen job

Use the following example as a model.

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header
'Accept: application/json' --header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' -d '{
  "unique_id": "my job",
  "type": "listen",
  "parameters": {
    "provider": "MyListenJob"
  }
}' 'https://localhost/1.0/rest-observer/jobs/listen'
```

Verify that the listen job is running

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X GET --header 'Accept: application/json' --header 'X-
TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' 'https://localhost/1.0/rest-
observer/jobs/my%20job'
```

Create a 'person' resource

This example creates a person resource called 'Thomas Watson'.

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header
'Accept: application/json' --header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --
header 'JobId: my job' -d '{
  "name": "Thomas Watson",
  "uniqueId": "Thomas Watson",
  "entityTypes": [
    "person"
  ]
}' 'https://localhost/1.0/rest-observer/rest/resources'
```

Create an 'organization' resource

This example creates an 'organization' resource of 'IBM'.

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header
'Accept: application/json' --header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --
header 'JobId: my job' -d '{
  "name": "IBM",
  "uniqueId": "IBM",
  "entityTypes": [
    "organization"
  ]
}' 'https://localhost/1.0/rest-observer/rest/resources'
```

Create a 'manages' relationship

This example creates a 'manages' releationship between Thomas Watson and IBM.

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header
'Accept: application/json' --header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --
header 'JobId: my job' -d '{
  "_fromUniqueId": "Thomas Watson",
```

```
  "_edgeType": "manages",
  "_toUniqueId": "IBM"
}' 'https://localhost/1.0/rest-observer/rest/references'
```

Create a new 'location' resource

This example creates a 'location' resource of 'Campbell, New York' for Thomas Watson, and a location relationship (an edgeType of locatedAt).

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header
'Accept: application/json' --header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --
header 'JobId: my job' -d '{
  "name": "Campbell, New York",
  "uniqueId": "Campbell, New York",
  "entityTypes": [
    "location"
  ],
  "_references": [
    {
      "_fromUniqueId": "Thomas Watson",
      "_edgeType": "locatedAt"
    }
  ]
}' 'https://localhost/1.0/rest-observer/rest/resources'
```

Replace the location resource with one having latitude and longitude properties

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X PUT --header 'Content-Type: application/json' --header
'Accept: application/json' --header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --
header 'JobId: my job' -d '{
  "name": "Campbell, New York",
  "uniqueId": "Campbell, New York",
  "entityTypes": [
    "location"
  ],
  "latitude": 42.232909,
  "longitude": -77.196918
}' 'https://localhost/1.0/rest-observer/rest/resources'
```

Delete all 'locatedAt' relationships

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X DELETE --header 'Accept: application/json' --header 'X-
TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --header 'JobId: my job'
'https://localhost/1.0/rest-observer/rest/resources/Thomas%20Watson/references/both/locatedAt'
```

Delete the location

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X DELETE --header 'Accept: application/json' --header 'X-
TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --header 'JobId: my job'
'https://localhost/1.0/rest-observer/rest/resources/Campbell%2C%20New%20York'
```

Delete the manages relationship

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X DELETE --header 'Accept: application/json' --header 'X-
TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --header 'JobId: my job'
'https://localhost/1.0/rest-
observer/rest/resources/Thomas%20Watson/references/both/manages/IBM'
```

Example bulk-replace job

The following examples show how to use the REST Observer bulk replace job to create and adjust a small topology.

Note: These examples use a mixture of the provided scripts in $ASM_HOME/bin and the cURL command.

Submit a bulk-replace job request with the unique ID of 'my bulk replace'

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header
'Accept: application/json' --header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' -d '{
  "unique_id": "my bulk replace",
  "type": "bulk replace",
  "parameters": {
    "provider": "Me"
  }
}' 'https://localhost/1.0/rest-observer/jobs/bulk_replace'
```

Verify that the bulk-replace job is running

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X GET --header 'Accept: application/json' --header 'X-
TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' 'https://localhost/1.0/rest-
observer/jobs/my%20bulk%20replace'
```

Submit a location resource for the city of Coventry

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header
'Accept: application/json' --header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --
header 'JobId: my bulk replace' -d '{
  "name": "Coventry",
  "uniqueId": "Coventry",
  "entityTypes": [
    "location"
  ]
}' 'https://localhost/1.0/rest-observer/rest/resources'
```

Submit a location resource for the town of Rugby

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header
'Accept: application/json' --header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --
header 'JobId: my bulk replace' -d '{
  "name": "Rugby",
  "uniqueId": "Rugby",
  "entityTypes": [
    "location"
  ]
}' 'https://localhost/1.0/rest-observer/rest/resources'
```

Submit a location resource for the Town of Leamington Spa with relationships to the existing resources of Coventry and Rugby

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header
'Accept: application/json' --header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --
header 'JobId: my bulk replace' -d '{
  "name": "Leamington Spa",
  "uniqueId": "Leamington Spa",
  "entityTypes": [
    "location"
  ],
  "_references": [
     {
        "_toUniqueId": "Coventry",
        "_edgeType": "routesVia"
     },
     {
        "_toUniqueId": "Rugby",
        "_edgeType": "routesVia"
     }
  ]
}' 'https://localhost/1.0/rest-observer/rest/resources'
```

Check your progress by rendering the topology

Having completed this set of observations, initiate a synchronize request for 'my bulk replace' job.

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header
'Accept: application/json' --header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' -d ''
```

```
'https://localhost/1.0/rest-observer/jobs/my%20bulk%20replace/synchronize'
```

Provide a new topology for the 'my bulk replace' job

This example submits a location resource for the town of Leamington Spa with relationships to the towns of Warwick and Stratford-Upon-Avon (resource placeholders).

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header
'Accept: application/json' --header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --
header 'JobId: my bulk replace' -d '{
  "name": "Leamington Spa",
  "uniqueId": "Leamington Spa",
  "entityTypes": [
    "location"
  ],
  "_references": [
     {
        "_toUniqueId": "Warwick",
        "_edgeType": "routesVia"
     },
     {
        "_toUniqueId": "Stratford-upon-Avon",
        "_edgeType": "routesVia"
     }
  ]
}' 'https://localhost/1.0/rest-observer/rest/resources'
```

Provide the resource data for the Town of Warwick resource placeholder, thus fully creating the resource

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header
'Accept: application/json' --header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' --
header 'JobId: my bulk replace' -d '{
  "name": "Warwick",
  "uniqueId": "Warwick",
  "entityTypes": [
    "location"
  ]
}' 'https://localhost/1.0/rest-observer/rest/resources'
```

Provide the resource data for the town of Stratford-upon-Avon resource placeholder, thus fully creating the resource

```
curl -u PROXY_USER[:PROXY_PASSWORD] -X POST --header 'Content-Type: application/json' --header
'Accept: application/json' --header 'X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255' -d ''
'https://localhost/1.0/rest-observer/jobs/my%20bulk%20replace/synchronize'
```

Initiate a synchronize request for the 'my bulk replace' job. This signifies to the Observer that it should instruct the system to replace the previous set of observations with the new ones.

Note: The new data is available immediately as it is provided. The synchronize request simply deletes any resources previously observed that were not observed this time. In the current example, Coventry and Rugby were not observed, and therefore they are deleted.

```
[root@asm-backend asm]# env bulk_replace_unique_id="my bulk replace"
./bin/rest_observer_bulk_replace_synchronize.sh
```

Configuring SDC ONAP Observer jobs

You configure SDC (Service Design and Creation) ONAP (Open Network Automation Platform) Observer jobs to dynamically load data from the SDC REST interface.

Before you begin

Important: The SDC ONAP Observer supports the SDC release 8.0.0.

Ensure you have the SDC ONAP service details to hand, such as username, password, hostname and port.

To obtain SDC ONAP system details:

1. To get port

   ```
   kubectl describe svc --selector=app.kubernetes.io/name=sdc-be | grep NodePort
   ```

2. To get message bus port

   ```
   kubectl describe svc --selector=app.kubernetes.io/name=message-router | grep NodePort
   ```

3. To copy trust store

   ```
   kubectl cp <onap-sdc-be-*>:/var/lib/jetty/chef-solo/environments/org.onap.sdc.trust.jks
   ./org.onap.sdc.trust.jks
   ```

   Or

   ```
   kubectl cp <onap-sdc-be-*>:/app/jetty/chef-solo/environments/org.onap.sdc.trust.jks
   ./org.onap.sdc.trust.jks
   ```

4. To get trust store password

   ```
   kubectl exec -it <onap-sdc-be-*> -- sh -c "cat /var/lib/jetty/chef-
   solo/environments/AUTO.json | grep truststore_pwd"
   ```

   Or

   ```
   kubectl exec -it <onap-sdc-be-*> -- sh -c "cat /app/jetty/chef-
   solo/environments/AUTO.json | grep truststore_pwd"
   ```

About this task

The SDC ONAP Observer job retrieves the service design data from the SDC server through REST. The observer loads and updates the resources and their relationships within the Agile Service Manager core topology service.

SDC manages two levels of assets:

Resource

A fundamental capability, implemented either entirely in software, or as software that interacts with a hardware device. Each Resource is a combination of one or more Virtual Function Components (VFCs), along with all the information necessary to instantiate, update, delete, and manage the Resource. A Resource also includes license-related information.

There are three kinds of resource:

- Infrastructure: Cloud resources, such as Compute, Storage, etc
- Network: Network connectivity functions & elements, for example a Virtual Network Function (VNF)
- Application: Features and capabilities of a software application, for example a load-balancing function=

Service

A well-formed object comprising one or more resources.

Service designers create services from resources, and include all of the information about the service needed to instantiate, update, delete, and manage the service.

You define and start the following jobs:

Load job

Run this job whenever you need SDC ONAP topology data refreshed.

By default, these jobs are one-off, transient jobs that perform a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) manually from the Observer UI, or schedule them to run at set times when configuring them.

Listen job

A long-running job that consumes relevant messages from the SDC DMaaP broker continually until it is explicitly stopped, or until the observer is stopped.

You should run this job after a load job has been completed.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Load job parameters

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
SDC Distribution Engine host	Specify the host name of the SDC Distribution Engine	Required
SDC Distribution Engine port	Specify the port number of the SDC Distribution Engine	Required
SDC username	Specify the user name to use when connecting to the SDC	Required
SDC password	Specify the password to use when connecting to the SDC	Required. Plain text password can be entered.
Target system Key Store file name	Specify the key store name that contains either the CA certificate or the SDC's public key	Required. For information on encrypting the contents of the key store file, see step 2 of the Configuring observer jobs security topic.
Target system Key Store password	Specify the password to access the key store file	Required. Plain text password can be entered.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job should run, and whether it should run at regular intervals. By default the job runs immediately, and only once. Optionally you can specify a future date and time for the job to run, and then set it to run at regular intervals after that.	Optional. Transient (one-off) jobs only. If you set a job schedule, the run intervals must be at least 90 seconds apart, and if you set them at less than 15 minutes, a warning is displayed, as the frequency can impact system performance.
Observer job description	Enter additional information to describe the job	Optional

Listen job parameters

Parameter	Action	Details

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
SDC Distribution Engine host	Specify the host name of the SDC Distribution Engine	Required
SDC Distribution Engine port	Specify the port number of the SDC Distribution Engine	Required
SDC username	Specify the user name to use when connecting to the SDC	Required
SDC password	Specify the password to use when connecting to the SDC	Required. Plain text password can be entered.
SDC environment name	Specify the environment name configured on the SDC, for example `TEST`, `PROD`, `AUTO`, etc	Required
DMaaP message-router address	Specify the DMaaP message bus addresses on which to listen for distribution events	Required
DMaaP message-router port	Specify the DMaaP message bus port on which to listen for distribution events	Required
Target system Key Store file name	Specify the key store name that contains either the CA certificate or the SDC's public key	Required. For information on encrypting the contents of the service account key file, see step 2 of the Configuring observer jobs security topic.
Target system Key Store password	Specify the password to access the key store file	Required. Plain text password can be entered.
DMaaP polling interval (in seconds)	Specify how often (in seconds) to poll the DMaaP cluster for new model events	Required. The default is 20s.
DMaaP polling timeout (in seconds)	Specify the timeout value (in seconds) when polling DMaaP for new model events	Optional. The default is 20s.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Observer job description	Enter additional information to describe the job	Optional

Configuring ServiceNow Observer jobs

Using the ServiceNow Observer job, you can retrieve the configuration management database (CMDB) data from ServiceNow through REST API, using basic authentication credentials.

Important: The ServiceNow Observer supports the cloud and SaaS ServiceNow versions `New York`, `Quebec`, `Paris`, `Tokyo`.

Before you begin

- **Important**: If you have any restrictive EgressFirewall in place, ensure that they are updated to allow for this inbound connection. For more information, see Create an EgressFirewall.
- **Important**: Ensure that your ServiceNow user account has the required roles to configure this observer and to access resources from ServiceNow. You need the following roles:
  - `rest_api_explorer`
  - `web_service_admin`
- Ensure that you have the ServiceNow service details to hand to complete this observer configuration, such as username, password, and URL.

To add a role to your user account, complete the following steps:

1. Log in to your **ServiceNow** instance using an admin account.
2. Go to **User Administration** and select **Users** from the menu.
3. Select to create or edit a user:
   - To create a user, click **New**.
   - To edit an existing user, search for, and then select, the user.
4. From the user's information tab, select the **Roles** tab, then click **Edit**.
5. Assign the required roles to the user.
6. Click **Save**, then **Update**.

For more detailed descriptions of available roles, go to **User Administration** and select the **Roles** page. You can also assign other roles with similar privileges to the non-admin account. To find other roles with similar privileges, navigate to **System Security** and select **Access Control (ACL)**.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

ServiceNow jobs retrieve configuration management database (CMDB) data from ServiceNow through REST API. The Observer loads and updates the resources and their relationships within the topology service.

By default, the observer discovers the following relationship tables:

- /api/now/table/cmdb_ci_service_discovered (services data)
- /api/now/table/cmdb_rel_ci (connectivity information for inventory data)
- /api/now/table/cmdb_ci_service_calculated (calculated service data)
- /api/now/table/cmdb_ci_query_based_service (query-based service data)

The observer supports all system-created or user-created child tables of cmdb_ci, as well as the following resource:

- /api/now/table/sys_user (user data)

ServiceNow job: By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered. You can also run these jobs (again) manually from the Observer UI, or schedule them to run at set times when configuring them. Run this job whenever you want to refresh the ServiceNow topology data.

Mapping of ServiceNow object types to topology entity types

Table. ServiceNow object and entity types

ServiceNow object types	Entity types
cmdb_ci, cmdb_ci_service_discovered	based on sys_class_name attribute
sys_user	person

Tip

Important: Specify **cmdb_ci_{name}** tables to discover using the **ServiceNow tables to be discovered** parameter.

Follow these steps to run the ServiceNow glide script that lists availabe **cmdb_ci** tables.

Note: You must have admin access to run the script.

1. Login to the ServiceNow instance as the admin user.
2. Go to **System Definition > Scripts - Background**.
3. Paste the following script into the text box:

```
var g = new GlideAggregate('cmdb_ci');
g.addAggregate('COUNT', 'sys_class_name')
g.orderByAggregate('COUNT', 'sys_class_name');
g.query();
while (g.next()) {
gs.info(g.sys_class_name + ': ' + g.getAggregate('COUNT', 'sys_class_name'));
}
```

4. Run the script.
5. ServiceNow will list all **cmdb_ci** tables in the environment. Sample output:

```
*** Script: cmdb_ci_spkg: 1767
*** Script: cmdb_ci_computer: 814
*** Script: cmdb_ci_service: 41
*** Script: cmdb_ci_printer: 28
*** Script: cmdb_ci_server: 19
*** Script: cmdb_ci_win_cluster_node: 14
*** Script: cmdb_ci_database: 13
*** Script: cmdb_ci_web_server: 9
*** Script: cmdb_ci_win_server: 8
*** Script: cmdb_ci_unix_server: 7
*** Script: cmdb_ci_ups: 6
*** Script: cmdb_ci_email_server: 5
*** Script: cmdb_ci_netgear: 5
*** Script: cmdb_ci_rack: 5
*** Script: cmdb_ci_zone: 5
*** Script: cmdb_ci_linux_server: 4
*** Script: cmdb_ci_win_cluster: 4
*** Script: cmdb_ci_cluster_node: 3
*** Script: cmdb_ci_ip_router: 3
*** Script: cmdb_ci_msd: 3
*** Script: cmdb_ci_network_adapter: 3
*** Script: cmdb_ci_aix_server: 2
*** Script: cmdb_ci_computer_room: 2
*** Script: cmdb_ci_disk: 2
*** Script: cmdb_ci_peripheral: 2
*** Script: cmdb_ci: 1
*** Script: cmdb_ci_appl: 1
*** Script: cmdb_ci_app_server_java: 1
*** Script: cmdb_ci_cluster: 1
*** Script: cmdb_ci_datacenter: 1
*** Script: cmdb_ci_db_mysql_catalog: 1
*** Script: cmdb_ci_db_ora_catalog: 1
*** Script: cmdb_ci_ip_switch: 1
*** Script: cmdb_ci_service_group: 1
*** Script: cmdb_ci_storage_switch: 1
```

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Remember: To improve performance and to focus your topology, **do not** include all ServiceNow tables. Instead (**before** running an observer job) identify the ServiceNow tables that are most useful to your requirements, and exclude the ones you

do not require. For example, you may want to exclude the vast number of records of SIM cards from being displayed in the topology, and include only servers, services or applications.

Table. Parameter action details

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
ServiceNow instance	Specify the ServiceNow instance	Required
ServiceNow username	Specify the ServiceNow username	Required
ServiceNow password	Specify the ServiceNow password	Required
ServiceNow tables to be discovered	The supported tables are **`sys_user`**, and all system-created or user-created child tables of **`cmdb_ci`**. Using a comma as delimiter for multiple entries, specify either **`sys_user`** or tables that are either system-created or user-created children of the **`cmdb_ci`** table, such as **`cmdb_ci_database`** (the system table), **`u_ibm_database`** (the global scope custom table), or **`sn_app_ws_u_ibm_database`** (the application scope custom table). **Note** that the following relationship tables are added by default: **`cmdb_rel_ci`**, **`cmdb_ci_service_discovered`**, **`cmdb_ci_service_calculated`**, and **`cmdb_ci_query_based_service`**.	Required
Number of records to fetch per API call from a table	Enter the number of records to fetch per API call from a table. The default value is 10000. The value you enter should be in the range of 100 to 10000.	Optional
Entity mapping for sysClassNames	This parameter will override the default observer mapping for the sys_class_name entity type with a user-defined entity type. Enter the class name and the new entity type name, where the values of the class and the new entity type name are separated by a colon. Example: **`cmdb_ci_linux_server:node`**.	Optional
Options to exclude entity edges	Select an exclude option to exclude resource edges	Optional
List of entities to exclude edge modelling	Enter the entities to exclude resource edges. The listed entities will be not have any edge connectivity to any entities	Optional
Use **`include`** or **`exclude`** filter	Specify the type of filter to use when retrieving resources. The options are **`include`** (default), or **`exclude`**. Specify whether to include or exclude an API filter in every ServiceNow API call.	Optional. If you select **`include`**, you specify the filter next.
Filter	Specify a filter using the JSON **`key:value`** attribute. The format used is **`key=value`**, where value can be more than one: **`key=value1,value2,value3`** For more than one key, use the vertical bar symbol (|) as delimiter: **`key=value1|key2=value1,value2,value3`**	Optional. Required only if the **`include`** option has been set.
Number of connection retry times	Set the number of connection retries.	Optional. The default is **`1`**.
Delay in (milliseconds) before a connection retry	Set the delay in milliseconds before a connection retry.	Optional. The default is **`5000ms`**.
Maximum number of tables	Enter the number of batch calls into which you want to split each API query. Defining a number greater than one (up to a maximum of 10) can avoid the query from being too long, thereby avoiding **`query too long`** ServiceNow API errors.	Optional. The default and maximum is 10.
Connection timeout (milliseconds)	Set the connection timeout in milliseconds.	Optional. The default is **`5000ms`**.

Parameter	Action	Details
Read timeout (milliseconds)	Set the read timeout in milliseconds.	Optional. The default is `5000ms`.
Proxy Host	Specify the proxy host to connect to.	Optional
Proxy Port	Specify the proxy port; defaults to `8080`.	Optional
Proxy Username	Specify the proxy user name to connect as. Set for basic auth proxy.	Optional
Trust all certificates by bypassing certificate verification	Set to true to allow connection to target environment without verification.	Optional. The default is 'false'.
Proxy Password	Specify the password for the proxy user (if `Proxy Username` has been specified).	Optional
Proxy Secure	Specify whether to connect to the proxy in secure mode. Set to True for HTTPS proxy.	Optional. The default is `false`.
Access scope	Enter text to provide a scope for the resources. Access scope can help map alerts to resources when resources in different scopes share the same parameters, such as matchTokens.	Optional. You can define access scope for locations, project names, namespaces, etc.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Set the time when the job should run, and whether it should run at regular intervals. By default, the job runs immediately, and only once. Optionally, you can specify a future date and time for the job to run, and then set it to run at regular intervals after that. The run intervals must be at least 90 seconds apart, and if you set them at less than 15 minutes, a warning is displayed, as the frequency can impact system performance.	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Configuring SevOne Observer jobs

Using the SevOne Observer functionality, you can load monitored servers and their relationships, and then visualize this data as a topology view.

Before you begin

Important: The SevOne Observer supports SevOne version 6.2.0 & API version 2.20.0.

Ensure you have the SevOne server details to hand, such as the username and password.

About this task

A SevOne Observer job extracts SevOne managed servers and their relationships via REST API calls.

You define and start the following job.

SevOne Load job

Run this job whenever you need SevOne topology data refreshed.

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

SevOne resources and relationships loaded by the SevOne Observer job

Vertex	Description
Server	SevOne managed server
Ipaddress	IP address associated with the server managed by SevOne

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

SevOne Observer job parameters for Load job

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Data Tenant	Specify the data tenant used for the device group or groups.	Required
Hostname	Enter the SevOne hostname	Required
Username	Specify the SevOne username	Required
Password	Specify the SevOne password	Required. Use plain text.
Device group path	Specify the device group path to be discovered. To discover more than one group, enter a comma-separated list.	Required
Number of SevOne hops	Choose the number of hops to be discovered	Required. Default is 1, maximum hops is 3.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Configuring Tivoli Application Dependency Discovery Manager Observer jobs

IBM Tivoli Application Dependency Discovery Manager Observer jobs retrieve network topology data, including discovered applications, their components, configurations and dependencies, from the Tivoli Application Dependency Discovery Manager database server (running either an IBM Db2 or an Oracle database), and use this data to create topology views.

Before you begin

Important: The Tivoli Application Dependency Discovery Manager Observer supports the on-premises Tivoli Application Dependency Discovery Manager version.

To connect to a Tivoli Application Dependency Discovery Manager Oracle database, you must place the Oracle JDBC Driver into the $ASM_HOME/lib folder, and then restart the observer for it to take effect. You can download the driver from the Oracle website, or copy it from the Oracle server (**not** the Oracle client) from the following location: ../app/oracle/product/Oracle_version/dbhome/jdbc/lib/ojdbc6.jar

Ensure you have the Tivoli Application Dependency Discovery Manager Rest API login access details in hand, such as the Tivoli Application Dependency Discovery Manager API URL, username and password.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The Tivoli Application Dependency Discovery Manager Observer is built on the Observer framework:

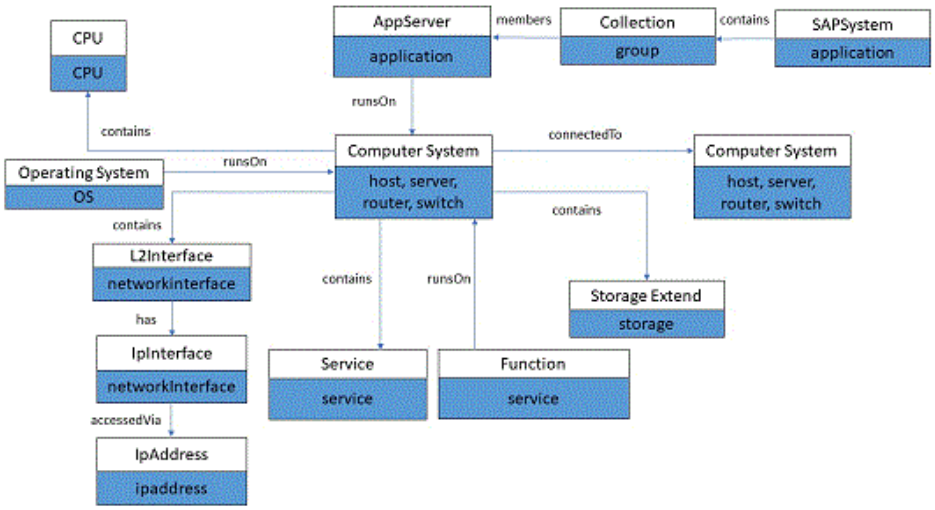


Figure. Observer framework

- A computer system can be a host, server, router, or switch
- A computer system contains CPU, L2Interface, and storage
- Operating system, application server and service run on a computer system
- A computer system can connect to another computer system
- A SAPSystem contains collection
- An application server can be a member of a collection

Mapping Tivoli Application Dependency Discovery Manager model objects to topology entity types

Tivoli Application Dependency Discovery Manager model object	Entity types
AppServer	`application`
ComputerSystem	`host`, `server`, `router`, `switch`
CPU	`cpu`
L2Interface	`networkinterface`
IpInterface	`networkinterface`
IpAddress	`ipaddress`
OperatingSystem	`os`
Service	`service`
StorageExtent	`storage`
Function	`service`
SAPSystem	`application`

Tivoli Application Dependency Discovery Manager model object	Entity types
Collection	`group`

Tivoli Application Dependency Discovery Manager Observer jobs retrieve topology data using the Tivoli Application Dependency Discovery Manager REST API, and the observer loads and updates the resources and their relationships within the topology service.

You define and start the following jobs.

Load job

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
TADDM API URL	Specify the Tivoli Application Dependency Discovery Manager endpoint to connect to	Required.
TADDM username	Specify the Tivoli Application Dependency Discovery Manager username	Required
TADDM password	Specify the password for the Tivoli Application Dependency Discovery Manager user	Required
TADDM objects to observe	Select one or more options from the drop-down list	Optional. If none are selected, all supported model objects are retrieved.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Configuring Viptela Observer jobs

You use the Viptela Observer to load managed Viptela SD-WAN control plane, vEdge and their associated network resources, and then visualize this data as a topology view.

Before you begin

Important: The Viptela Observer supports version 18.4.302 of Cisco SD-WAN.

Ensure you have the server details to hand, such as the username and password.

About this task

The Viptela Observer targets the Cisco SD-WAN (or 'Cisco Viptela') system. A Viptela Observer job extracts server information and its associated network resources from Viptela through REST API. The observer loads and updates the resources and their relationships and then displays them as a topologgy view.

You define and start the following job.

Full Topology Upload job

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Data center name	Specify the data center the vEdge runs in.	Required. If you want to discover multiple vEdge locations or sets, then provide a different data center, for example `EURDC01,USDC01`.
Cisco Viptela vManage host IP	Enter the Cisco SD-Wan Viptela vManage host IP.	Required
Cisco Viptela API port.	Specify the Cisco SD-Wan Viptela vManage API port.	Required. The default value is 8443.
Username to login to Cisco Viptela service	Specify the Cisco SD-Wan Viptela vManage username.	Required
Password for Cisco Viptela user	Specify the Cisco SD-Wan Viptela vManage password.	Required. A plain text password can be entered.
Include the control plane data	Select true to load the control plane data.	Optional
vEdge grouping preference for defined topology	Specify the vEdge grouping preference for your topology.	Optional. Supported grouping: `device-group (default)`, `site-id Data Tenant`
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job.	Optional

Configuring VMware NSX Observer jobs

You configure VMware NSX Observer jobs to dynamically load data from the VMware NSX REST interface. You can use the VMware NSX Observer when you have a VMware NSX appliance in your environment.

Before you begin

Important: The VMware NSX Observer supports the on-premises VMware NSX version 6.3.

Ensure you have the VMware NSX REST API details to hand, such as the VMware NSX URL, username, password, and SSL truststore.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The VMware NSX Observer job extracts VMware NSX resource information through REST, and the observer loads and updates the resources and their relationships within the topology service.

You define and start the following job.

VMware NSX Observer job (full topology load) This job loads a baseline of topology data from an environment, which contains a VMware NSX appliance.

Run this job whenever you need VMware NSX topology data refreshed.

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

The VMware NSX Observer loads the following resources and their relationship into the IBM Cloud Pak® for AIOps core topology service:

- NSX Appliance
- vCenter Appliance
- NSX Controller
- Edge Router - Logical (Distributed) Router, Edge Service Gateway
- Virtual machines
- Host
- VNIC

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
Network Virtualization and Security Platform password	Enter the password to authenticate with	Required

Parameter	Action	Details
Network Virtualization and Security Platform API URL	Specify the API URL of the VMware NSX endpoint	Required. Usually in the following format: https://[hostname or IP address]/api
SSL truststore file	Specify the truststore file name	Required
SSL truststore file password	Specify the truststore password to decrypt the HTTPS truststore file	Required.
SSL Validation	Choose whether SSL validation is on or off. Turning SSL validation off will use HTTPS without host verification.	Optional
Connection and read timeout (ms)	Enter the time at which the connection and read actions time out	Optional. Must be a value greater than 0 (zero), and the default is 5000 (5 seconds).
Data center name	Specify the data center in which the VMware NSX instance runs	Required.
Network Virtualization and Security Platform username	Specify the username to connect as, or listen to	Required
Network Virtualization and Security Platform tenant name	Specify the tenant	Optional.
Network Virtualization and Security Platform certificate	Specify a certificate by name to load into the truststore	Required.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Configuring VMware vCenter Observer jobs

You configure VMware vCenter Observer jobs to dynamically load data from the VMware vCenter REST interface.

Before you begin

Important: The VMware vCenter Observer supports the on-premise VMware vCenter versions 6.5, 6.7 and 7.0.3.

Ensure you have the VMware vCenter service details to hand, such as username, password, SSL truststore and URL.

Important: If you have any restrictive EgressNetworkPolicies in place, ensure that they are updated to allow for this inbound connection. For more information, see Postinstallation tasks.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

The VMware vCenter Observer job extracts VMware vCenter resource information through REST, and the observer loads and updates the resources and their relationships within the topology service.

You define and start the following job.

VMware vCenter Observer job (full topology load)

Run this job whenever you need VMware vCenter topology data refreshed.

By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

The VMware vCenter Observer loads the following resources and their relationship into the topology service:

- ESXi / ESX Hosts
- Virtual machines
- VNICs
- Storage

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
vCenter password	Enter the password to authenticate with	Required
vCenter API URL	Specify the API URL of the VMware vCenter endpoint (including port and version)	Required. Usually in the following format: https://[hostname or IP address]/rest
HTTPS truststore file name	Specify the truststore file name. The observer generates the trust store file based on the file name provided. **Tip:** You can use the observer name as file name (`<observer>.jks`), for example `VMVCenter.jks`.	Required. The supported format is JKS and the file is relative to $ASM_HOME/security.
truststore file password	Specify the truststore password the observer will use to decrypt the HTTPS truststore file. Use a password that conforms to your internal security requirements.	Required

Parameter	Action	Details
Data center name	Specify the data center in which the VMware vCenter instance runs	Required. If more than one, list them (comma-separated).
vCenter username	Specify the username to connect as, or listen to	Required
vCenter certificate	Specify a certificate by name to load into the truststore	Required
Certificate validation	Choose whether SSL validation is on or off. Turning SSL validation off will use HTTPS without host verification.	Optional
Connection and read timeout	Enter the time at which the connection and read actions time out	Optional. Must be a value greater than 0 (zero), and the default is 5000 (5 seconds).
vCenter host's name regex to discover	Specify an exact match or a regular expression match for a host's name in order to discover all its virtual machines	Optional
Connection retry times	Set the connection retry times	Optional. The default value is 5.
Retry after delay (milliseconds)	Set the time delay before trying to reconnect (in milliseconds)	Optional. The default value is 1000.
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Configuring Zabbix Observer jobs

Using the Zabbix Observer functionality, you can load monitored servers and their associated network resources, and then visualize this data as a topology view.

Before you begin

The Zabbix Observer supports Zabbix 4.0.3 and 6.0.3.

Ensure you have the Zabbix server details to hand, such as the username, password and SSL truststore.

Enabling access to the URL routes

To access the URL routes for the topology Swagger documentation, see the Enabling access to URL routes topic.

About this task

A Zabbix Observer job extracts server information and its associated network resources from Zabbix through REST RPC, and the observer loads and updates the resources and their relationships within the topology service.

You define and start the following job.

Full Topology Upload job By default, Load jobs are one-off, transient jobs that do a full upload of all requested topology data as soon as they are triggered.

You can also run these jobs (again) from the Observer UI, or schedule them to run at set times when configuring them.

Procedure

Define or edit the following parameters, then click **Run job** to save and run the job.

Encryption requirement: See the Configuring observer jobs security topic for more information.

Parameter	Action	Details
Unique ID	Enter a unique name for the job	Required
The datacenter of zabbix server	Specify the data center in which the Zabbix instance runs	Required. If more than one, list them (comma-separated).
The hostname of zabbix server	Enter the Zabbix virtual hostname	Required
The username of zabbix server	Specify the Zabbix username	Required
The password of zabbix server	Specify the Zabbix user password	Required
Zabbix SSL certificate	Specify a certificate file name	Required
SSL Validation	Choose whether to use SSL validation ('true' or 'false'). If false, HTTPS is used, but without hostname validation.	Optional
HTTPS truststore file name	Specify the truststore file name.	Required
truststore file password	Specify the truststore password to decrypt the HTTPS truststore file	Required
Zabbix connection timeout (milliseconds) (optional)	Timeout, in ms, when querying the Zabbix REST API	Optional. Default is 5000 (5s).
Generate debug support file	Set this parameter to 'True' in order to capture the output of the next scheduled job run as a file. This file will be stored with an observer's log files and can be used to debug observer issues, for example at the request of your designated Support team, or while using a test environment. For one-off jobs (that is, Load jobs), this parameter reverts to 'False' after the next completed run. To examine the output produced, you can load the generated debug file using the File Observer.	Optional
Job schedule	Specify when the job runs	Optional. Load jobs only.
Observer job description	Enter additional information to describe the job	Optional

Configuring observer job security

Some observers require authentication credentials such as certificates, keystores, or both. This topic describes such configuration tasks.

Note: Information on specific observers is located in the subtopics under Defining observer jobs.

To configure the security, complete the following steps:

- Obtain an authentication certificate.
- Store that certificate as a secret.

Important: Only update **`topology-custom-secrets`** with the Red Hat OpenShift Container Platform command-line interface (CLI). If you update custom secrets in the console, the crypto key is corrupted and the command to encrypt passwords does not work.

Managing authentication certificates and storing them as secrets (Red Hat OpenShift Container Platform)

1. Obtain the authentication certificate by using OpenSSL.

   ```
   echo -n | openssl s_client -connect {ipAddress}:{port} | sed -ne '/-BEGIN
   CERTIFICATE-/,/-END CERTIFICATE-/p' | base64 -w 0 > target_system.crt
   ```

   Where **`target_system.crt`** contains the encoded certificate, and **`{ipAddress}`** is the IP address of any of the following target systems:

   - Ansible
   - Big Cloud fabric
   - BigFix Inventory
   - Ciena Blue Planet
   - Cisco ACI
   - Datadog
   - Dynatrace
   - Juniper CSO
   - HPNFVD
   - GitLab
   - Kubernetes master node
   - OpenStack
   - Rancher
   - SDC ONAP
   - VMware NSX
   - VMware vCenter
   - Zabbix

   Example `target_system.crt`:

   ```
   [root@localhost ~]# cat target_system.crt
   LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUN3RENDQWFnQ0NRRGRuMENqU3BXZXhUQU5CZ2txaGtpRzl3M
   EJBUVVGQURBaU1RMHdDd1lEVlFRRERBUkIKVUVsRE1SRXdEd1lKS29aSWh2Y05BUWtCRmdKVlV6QWVGdzB4TmpBeE
   1qRXdOekV5TWpWYUZ3MHlOakF4TVRndwpOekV5TWpWYU1DSXhEVEFMQmdOVkJBTU1CRUZRU1VNeEVUQVBCZ2txaGt
   pRzl3MEJDUUVXQWxWVE1JSUJJakFOCkJna3Foa2lHOXcwQkFRUUZBQU9DQVE4QU1JSUJDZ0tDQVFFQW1Ob0dxdFI0
   RlFPWkdoUWFtand1YmxRYjRobU0KTzJwOGtjbGUwL2NuUno3cSttWGYzWlRQYTZsWEk2MG9BbmVPSGowZEVaMkhwR
   WFFblBUbWJmWUF6Y0ZQdjNVWApMWjM3VWVoMDZXTjMxS29tSSs2czJtSk1IWWM0MW44M1RiUU5uWUNjYjZjdlZLcW
   V5NVhhaFBtdkZDbDBtM3Y3Cisxa1lFMFRNVlBnTk56R0ZSUXU1RVlGc3FZWHZGbFZhZ0lUalF6cks3YnE0RkJiMW1
   kVjFsYnVOMWhISzd2SFEKS3ZUNHBGbGx1NTRHU0JhZ2RSbUdad0dta0tNZHRGUkEvc3pBWEMrejQ0cHN3T05ydTJn
   bDR3bG5MZTVvM2NWZwpFQUx1THM4UDgrOUx0eFN3YWJvb0VMcHRjb3pKdEpUb2E4QS9zZXRaSi81RUJQNmhjNk1yU
   WxHQktRSURBUUFCCk1BMEdDU3FHU0liM0RRRUJCUVVBQTRJQkFRQkJuZzlJK2pBdjhNUjBYemM1SUUxdTBkK0JweW
   9OZGVRbkl4T2sKZWFsNzNUbmkzWmh4QUQzdlQzenZNSElSUEc0d3lxMWJqQ05LY3BZOGVCbVJuVzhOSnlNdG9vcU9
   hNlJMWGNPTAoyeVZublVna092THRPVjM5eFNFQ1BOMzV4YXJJdGYydE9NZWJRWlc1ZC9Hc1lPZUFLTlNrTlQwRmtr
   eDE0UzJFClpBVi9IUUVHaVpUR0tQNkx1czYzLzJiTEJVNHdGUjg3bjNkdFJFVUp5eGQ4ZDJDTFA4MkE2UTNOT2lOZ
   Edkam0KSnFQZXNEaWxXWE5Gd09xUk1XOWFGWTVUSUtOL25PQzhqczI0cVFmZTJZcllnZ242N0crLytBQy9kV21JSV
   Q2dgpBWTVMejhwOWQwSzZUaGxLeVpNZkdYVkNnMFlvTms1ajQ4ckJlZ2J5cFhTM1J2SnIKLS0tLS1FTkQgQ0VSVEl
   GSUNBVEUtLS0tLQo=

   [root@locahost ~]#
   ```

2. **For the SDC ONAP Observer**, encrypt the contents of the key store file using base64.

For example:

```
cat {onap_sdc_file_name}.jks | base64 -w 0 > org.onap.sdc.trust.txt
```

Where **org.onap.sdc.trust.txt** contains the encoded key store file, and **{onap_sdc_file_name}.jks** is the key store file downloaded from the target ONAP SDC system.

Next, store the encoded file as a secret.

3. **For the GoogleCloud Observer**, encrypt the contents of the service account key file by using base64.

   For example:

   ```
   cat {project_id_file_name}.json | base64 -w 0 > googlecloud.json
   ```

   Where **googlecloud.json** contains the encoded service account key file, and **{project_id_file_name}.json** is the service account key file you downloaded from **Credentials** under **API & Services** in the Google Cloud Platform dashboard.

   Next, store the encoded service account key file as a secret.

4. Store the encoded certificate that was generated in the previous step as a secret.

   Each installed IBM Cloud Pak for AIOps release has a single special secrets file. Data added to that is made available to the appropriate observer containers.

   - Find the correct secrets file by using the following command:

     ```
     oc get secret <releasename>-custom-secrets
     ```

     Where the value of **<releasename>** and the secret can be obtained by using the following example:

     ```
     oc get asm
     ```

     Sample output:

     ```
     NAME             AGE
     aiops-topology   12d
     ```

     Find the secret for **aiops-topology**:

     ```
     oc get secret aiops-topology-custom-secrets
     ```

     Sample output:

     ```
     NAME                              TYPE     DATA   AGE
     aiops-topology-custom-secrets     Opaque   4      12d
     ```

   - Edit the appropriate file for your release.

     ```
     oc edit secret <releasename>-custom-secrets
     ```

   - Add a name and value pair to the data section.

     The value is the certificate generated earlier. The name is the value that you entered as the certificate file name to run the observer job.

     Example: The following sample shows the expected content in the secret file after adding **vcenter.crt**. The data section is between the **apiVersion** and **kind** sections.

     ```
     data:
     {name}:{value}
     ```

     Note: This VMware vCenter Observer example registers the **vcenter.crt** SSL certificate in Red Hat OpenShift Container Platform Secret. **vcenter.crt** is the job parameter value for the VMware vCenter Observer. Define a new **{name}** parameter in the same file for other observers that require SSL certificates. You provide the certificate in Red Hat OpenShift Container Platform Secret and the settings you provide for truststore/truststore password is used to generate a new truststore automatically.

```
apiVersion: v1
data:
vcenter.crt:LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUN3RENDQWFnQ0NRRGRuMENqU3BXZXhU
QU5CZ2txaGtpRzl3MEJBUVVGQURBaU1RMHdDd1lEVlFRERBUkIKVUVsRE1SRXdEd1lKS29aSWh2Y05BUWtC
RmdKVlV6QWVGdzB4TmpBeE1qRXdOekV5TWpWYUZ3MHlOakF4TVRndwpOekV5TWpWYU1DSXhEVEFMQmdOVkJB
TU1CRUZRU1VNeEVUQVBCZ2txaGtpRzl3MEJDUUVXQWxWVE1JSUJJakFOCkJna3Foa2lHOXcwQkFRUZBQU9D
QVE4QU1JSUJDZ0tDQVFFQW1Ob0dxdFI0RlFPWkdoUWFtand1YmxRYjRobU0KTzJwOGtjbGUwL2NuUno3cStt
WGYzWlRQYTZsWEk2MG9BbmVPSGowZEVaMkhwRWFFblBUbWJmWUF6Y0ZQdjNVWApMWjM3VWVoMDZXTjMxS29t
SSs2czJtSk1IWWM0MW44M1RiUU5uWUNjYjZjdlZLcWV5NVhhaFBtdkZDbDBtM3Y3Cisxa1lFMFRNVlBnTk56
R0ZSUXU1RVlGc3FZWHZGbFZhZ0lUalF6cks3YnE0RkJiMW1kVjFsYnVOMWhISzd2SFEKS3ZUNHBGbGx1NTRH
U0JhZ2RSbUdad0dta0tNZHRGUkEvc3pBWEMrejQ0cHN3T05ydTJnbDR3bG5MZTVvM2NWZwpFQUx1THM4UDgr
OUx0eFN3YWJvb0VMcHRjb3pKdEpUb2E4QS9zZXRaSi81RUJQNmhjNk1yUWxHQktRSURBUUFCCk1BMEdDU3FH
U0liM0RRRUJCUVVBQTRJQkFRQkJuZzlJK2pBdjhNUjBYemM1SUUxdTBkK0JweW9OZGVRbkl4T2sKZWFsNzNU
bmkzWmh4QUQzdlQzenZNSElSUEc0d3lxMWJqQ05LY3BZOGVCbVJuVzhOSnlNdG9vcU9hNlJMWGNPTAoyeVZu
blVna092THRPVjM5eFNFQ1BOMzV4YXJJdGYydE9NZWJRWlc1ZC9Hc1lPZUFLTlNrTlQwRmtreDE0UzJFClpB
Vi9IUUVHaVpUR0tQNkx1czYzLzJiTEJVNHdGUjg3bjNkdFJFVUp5eGQ4ZDJDTFA4MkE2UTNOT2lOZEdkam0K
SnFQZXNEaWxXWE5Gd09xUk1XOWFGWTVUSUtOL25PQzhqczI0cVFmZTJZcllnZ242N0crLytBQy9kV21JSVQ2
dgpBWTVMejhwOWQwSzZUaGxLeVpNZkdYVkNnMFlvTms1ajQ4ckJlZ2J5cFhTM1J2SnIKLS0tLS1FTkQgQ0VS
VElGSUNBVEUtLS0tLQo=
kind:Secret
```

If the edit is successful, the following message will be displayed:

```
secret "<release-name>-custom-secrets" edited
```

5. In the Red Hat OpenShift Container Platform web console, you can view the configured secret under the **Menu** > **Workload** > **Secrets** option, where the **Name** is `'<releasename>-custom-secrets'`. Within `'<releasename>-custom-secrets'`, all data configured earlier displays.

Event Manager integrations

Probes connect to an event source, detect and acquire event data, and forward the data as events. They use the logic specified in a rules file to manipulate the event elements before converting them into fields of an event in the ObjectServer alerts.status table.

Note: You can connect IBM Netcool Operations Insight probes directly to IBM Cloud Pak for AIOps without needing to install the Netcool OMNIbus ObjectServer. If you do not have an existing version of Netcool or NOI deployed, install the probe on either a Linux VM or OCP (if supported), and follow the instructions in Installing the IBM Netcool Event Integrations Operator and common probe configuration steps to get the information of the AIOps ObjectServer and point the probe to the AIOps ObjectServer.

They can acquire data from any stable data source, including devices, databases, and log files. Some probes are uniquely designed to acquire event data from a specific source, while other more generic probes (for example, the Message Bus Probe) can be configured to integrate with a variety of sources.

- IBM® Netcool® Operations Insight® (on-premises) probe integration
- IBM SevOne NPM probe integration
- IBM Turbonomic probe integration
- Metric Manager probe integration
- Probe Common Tasks

Connecting with on-premises probes

If you want to connect stand-alone on-premises probes to IBM Cloud Pak for AIOps, you can use NodePorts to securely connect to the in-cluster ObjectServer.

These instructions can be used to connect an on-premises deployment of IBM® Netcool® Operations Insight® to IBM Cloud Pak for AIOps.

Complete the steps in both of the following topics.

- Configuring the Cloud system
- Configuring the on-premises system

Note: IBM Cloud Pak for AIOps does not support connecting with a uni-directional gateway.

Configuring the Cloud system

1. Run the following commands to expose the ObjectServer services as node ports.

Ensure that you replace **<name>** with the name of your IBM Cloud Pak® for AIOps instance.

For more information about the configured IBM Cloud Pak for AIOps ports, see Ports.

```
cat << EOF | oc apply -f -
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/component: ncoprimary
    app.kubernetes.io/instance: aiops
    app.kubernetes.io/managed-by: ir-core-operator
    app.kubernetes.io/name: IssueResolutionCore
  name: aiops-ir-core-ncoprimary-np
spec:
  ports:
  - name: tds
    port: 4100
    protocol: TCP
    targetPort: 4100
  selector:
    app.kubernetes.io/component: ncoprimary
    app.kubernetes.io/instance: aiops
    app.kubernetes.io/managed-by: ir-core-operator
    app.kubernetes.io/name: IssueResolutionCore
  sessionAffinity: None
  type: NodePort
EOF

cat << EOF | oc apply -f -
apiVersion: v1
kind: Service
metadata:
  labels:
    app.kubernetes.io/component: ncobackup
    app.kubernetes.io/instance: aiops
    app.kubernetes.io/managed-by: ir-core-operator
    app.kubernetes.io/name: IssueResolutionCore
  name: aiops-ir-core-ncobackup-np
spec:
  ports:
  - name: tds
    port: 4100
    protocol: TCP
    targetPort: 4100
  selector:
    app.kubernetes.io/component: ncobackup
    app.kubernetes.io/instance: aiops
    app.kubernetes.io/managed-by: ir-core-operator
    app.kubernetes.io/name: IssueResolutionCore
  sessionAffinity: None
  type: NodePort
EOF
```

2. Get the service node port for the primary:

```
NCOPRIMARY_NODEPORT=$(oc get svc aiops-ir-core-ncoprimary-np -o jsonpath='{.spec.ports[?
(@.port==4100)].nodePort}')
```

3. Get the service node port for the backup:

```
NCOBACKUP_NODEPORT=$(oc get svc aiops-ir-core-ncobackup-np -o jsonpath='{.spec.ports[?
(@.port==4100)].nodePort}')
```

4. Get the node internal IP addresses:

```
oc get node -o jsonpath='{.items[*].status.addresses[?(@.type=="InternalIP")].address}'
```

   The output of the command returns the master node IP addresses and worker node IP addresses. You can run the following command to confirm which IP address belongs to which node.

```
oc get nodes -o wide | grep <NAME_OF_MASTER_OR_WORKER_NODE>
```

   Where **<NAME_OF_MASTER_OR_WORKER_NODE>** is the name of the master node or worker node. For example, the name can be **master1** or **worker1**.

5. Depending on the configuration of your cluster, you might need to add network policies to allow network traffic from outside the cluster. To test this, use the commands below. If you are using a proxy such as HA proxy (see below) you should run the test on the host running the proxy. Otherwise, run the test on the host running your on-premise system. Each of the following commands should output **CONNECTED**, followed by a **Certificate chain** with the first subject containing **ncoprimary** or **ncobackup**.

```
openssl s_client -connect <WORKER0_IP>:${NCOPRIMARY_NODEPORT} < /dev/null
openssl s_client -connect <WORKER0_IP>:${NCOBACKUP_NODEPORT} < /dev/null
```

   Where

   - **<WORKER0_IP>** is the value of worker IP address from the node internal IP addresses command.
   - **${NCOPRIMARY_NODEPORT}** is the service node port for the primary ObjectServer.
   - **${NCOBACKUP_NODEPORT}** is the service node port for the backup ObjectServer.

6. If the test in step **5** fails, you might need to add network policies. The ingress "from" rules are specific to your cluster and the clients or proxies that you want to connect to the ObjectServers. The rest of the policies should look as follows:

```
---
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: ncoprimary-external-ingress
spec:
  ingress:
  - from:
      # Add ingress rules required for your cluster and on-premises system here
    ports:
    - port: 4100
      protocol: TCP
  podSelector:
    matchLabels:
      app.kubernetes.io/component: ncoprimary
      app.kubernetes.io/instance: aiops
  policyTypes:
  - Ingress
...
---
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: ncobackup-external-ingress
spec:
  ingress:
  - from:
      # Add ingress rules required for your cluster and on-premises system here
    ports:
    - port: 4100
      protocol: TCP
  podSelector:
    matchLabels:
      app.kubernetes.io/component: ncobackup
      app.kubernetes.io/instance: aiops
  policyTypes:
```

```
  - Ingress
...
```

7. Next, you must update any proxies and firewalls to ensure that the exposed node ports are reachable. For example, if you have a HA proxy in front of your Red Hat OpenShift Container Platform cluster, you only need to update the HA proxy configuration that is running on the main control node. This routes that port to the master and worker nodes in the cluster. When you are logged on to that cluster, perform the following steps:

    1. Run the command:

       ```
       systemctl stop haproxy.service
       ```

    2. Add the following entries to **/etc/haproxy/haproxy.cfg**:

       ```
       frontend ingress-ncoprimary
           bind *:<NCOPRIMARY_NODEPORT>
           default_backend ingress-ncoprimary
           mode tcp
           option tcplog

       backend ingress-ncoprimary
           balance source
           mode tcp
           server master0 <MASTER0_IP>:<NCOPRIMARY_NODEPORT> check
           server master1 <MASTER1_IP>:<NCOPRIMARY_NODEPORT> check
           server master2 <MASTER2_IP>:<NCOPRIMARY_NODEPORT> check
           server worker0 <WORKER0_IP>:<NCOPRIMARY_NODEPORT> check
           server worker1 <WORKER1_IP>:<NCOPRIMARY_NODEPORT> check
           server worker2 <WORKER2_IP>:<NCOPRIMARY_NODEPORT> check

       frontend ingress-ncobackup
           bind *:<NCOBACKUP_NODEPORT>
           default_backend ingress-ncobackup
           mode tcp
           option tcplog

       backend ingress-ncobackup
           balance source
           mode tcp
           server master0 <MASTER0_IP>:<NCOBACKUP_NODEPORT> check
           server master1 <MASTER1_IP>:<NCOBACKUP_NODEPORT> check
           server master2 <MASTER2_IP>:<NCOBACKUP_NODEPORT> check
           server worker0 <WORKER0_IP>:<NCOBACKUP_NODEPORT> check
           server worker1 <WORKER1_IP>:<NCOBACKUP_NODEPORT> check
           server worker2 <WORKER2_IP>:<NCOBACKUP_NODEPORT> check
       ```

       Where

       - **<MASTER0_IP>**, **<MASTER1_IP>**, and **<MASTER2_IP>** are the values of master IP addresses from node internal IP addresses command.
       - **<WORKER0_IP>**, **<WORKER1_IP>**, and **<WORKER2_IP>** are the values of worker IP addresses from node internal IP addresses command.
       - **<NCOPRIMARY_NODEPORT>** is the value of service node port for the primary ObjectServer.
       - **<NCOBACKUP_NODEPORT>** is the value of service node port for the backup ObjectServer.

    3. Run the command:

       ```
       systemctl start haproxy.service
       ```

8. Run the following command to retrieve the challenge credentials for the probe:

   ```
   oc get secret aiops-ir-core-omni-secret -o=jsonpath='{.data.OMNIBUS_PROBE_PASSWORD}' |
   base64 -d
   ```

Configuring the on-premises system

1. Get the service CA that signed the ObjectServer certificates:

Note: You must run this step on the cloud side and copy over the `service-ca.crt` to the on-premises server.

```
oc get configmap <name>-ir-core-service-ca -o jsonpath='{.data.service-ca\.crt}' >
service-ca.crt
```

2. Import the CA certificate into the IBM Tivoli Netcool/OMNIbus keystore:

```
$NCHOME/bin/nc_gskcmd -keydb -create -db $NCHOME/etc/security/keys/omni.kdb  -pw
"$KEYSTORE_PW" -stash
```

```
$NCHOME/bin/nc_gskcmd -cert -add -db $NCHOME/etc/security/keys/omni.kdb -pw
"$KEYSTORE_PW" -label "CA" -file service-ca.crt
```

Note: The default for `$NCHOME is /opt/IBM/tivoli/netcool/`.

3. Add the local host resolution for service name that is used in the CN of certificates to the IP of the cluster control node. This is so that the client can validate the certificate chain by the hostname used.

```
/etc/hosts:
<cluster_control_node_ip> -ir-core-ncoprimary..svc
<cluster_control_node_ip> -ir-core-ncobackup..svc
```

4. Update the `omni.dat` (`$NCHOME/etc/omni.dat`) with the necessary connection details, for example:

```
[AIOPS_AGGV]
{
        Primary: <name>-ir-core-ncoprimary.<namespace>.svc ssl ${NCOPRIMARY_NODEPORT)
        Backup: <name>-ir-core-ncobackup.<namespace>.svc ssl ${NCOBACKUP_NODEPORT)
}
```

Then, regenerate the interfaces file through `$NCHOME/bin/nco_igen`.

5. Try pinging the server by name:

```
$NCHOME/omnibus/bin/nco_ping AIOPS_AGGV
```

6. Specify the challenge credentials in the probe property files:

```
AuthUsername: "aiopsprobe"
AuthPassword: "<password from step 6 of on cloud side>"
```

Probe for IBM SevOne Network Performance Management (NPM) integration

You can configure the Probe integration for IBM SevOne Network Performance Management for connecting to IBM Cloud Pak for AIOps.

Note: Since the release of IBM Cloud Pak AIOps 4.1, you can now use the Generic Webhook Connector to integrate IBM Cloud Pak for AIOps with IBM SevOne Network Performance Management (NPM) to digest events. Using the Webhook Connector is preferred over installing a probe to connect with IBM SevOne as it simplifies the setup process. For details see Creating Generic Webhook connections.

- Overview
- Deployment Scenarios
- Installation

Overview

The Probe integration for IBM SevOne NPM creates a webhook endpoint to receive policy based alerts from IBM SevOne NPM. This integration is provided by the IBM Netcool Operations Insight Event Integrations Operator 3.4. Depending on your deployment you can configure the probe integration for IBM SevOne NPM connection to the IBM Cloud Pak for AIOps. If it is the IBM Cloud Pak for AIOps, the Probe integration for IBM SevOne NPM with IBM Cloud Pak for AIOps consists of the

following steps. As a prerequisite you need the following systems: IBM Cloud Pak for AIOps in openshift container platform, and IBM SevOne Network Management System (NMS) 6.1. You then need to install the IBM Netcool event integrations operator, gather IBM Cloud Pak for AIOps ObjectServer connection information, configure and install the probe for SevOne integration, obtain the probe Webhook URL, and then lastly, configure IBM SevOne Network Management System (NMS) to forward events to the probe for SevOne.

Deployment Scenarios

Depending on your deployment scenario, you can configure the Probe integration for IBM SevOne NPM connecting to the IBM Cloud Pak for AIOps.

Installation

Note: Using a generic webhook connection is preferred over installing a probe to connect with IBM SevOne, so you should consider migrating to this method. See Creating Generic Webhook connections.

The instructions to configure the probe with IBM Cloud Pak for AIOps are provided through their respective links below, and you should follow the instructions according to your deployment architecture.

IBM SevOne NPM Integration with IBM Cloud Pak for AIOps

Install the Probe integration for IBM SevOne NPM connecting to IBM Cloud Pak for AIOps.

Probe integration for IBM SevOne Network Performance Management (NPM) with IBM Cloud Pak for AIOps

You can use a probe to integrate IBM Cloud Pak for AIOps with IBM SevOne Network Performance Management (NPM).

When this integration is created, alert changes from SevOne are reported within IBM Cloud Pak for AIOps. Changes in IBM Cloud Pak for AIOps are not reported in SevOne.

The properties of the alerts include fields that are based on the IBM Netcool Operations Insight **`alert.status`** table columns (**`Identifier`**, **`AlertGroup`**, **`AlertKey`**, **`EventId`**, and more).

The following table shows comparative severity mapping:

SevOne Severity	IBM Netcool Operations Insight Severity	IBM Cloud Pak for AIOps Severity
Emergency, Alert, Critical	Critical	Critical
Error	Major	Major
Warning	Warning	Warning
Notice, Info, Debug	Informational	Informational
Indeterminate	Indeterminate	Indeterminate
Cleared (from clear condition)	(Creates a resolution event)	(Clears alert)

Prerequisites

You need to install the IBM Netcool Event Integrations Operator and gather the ObjectServer connection information before proceeding with defining the probe integration. For more information about installing the operator, see Installing the IBM Netcool Event Integrations Operator and common probe configuration steps

Configure and Install Probe For SevOne Integration

1. Set the probe namespace variable to the namespace where you want to install the probe. Usually this namespace is the same namespace as the IBM Netcool Operations Insight Event Integrations Operator.

```
PROBE_NAMESPACE=<probe-namespace>
```

2. Create a secret with the ObjectServer credentials for the probe to authenticate with the ObjectServer.

```
 PROBE_OMNI_SECRET=noi-probe-secret
 oc create secret generic $PROBE_OMNI_SECRET --namespace $PROBE_NAMESPACE \
  --from-literal=AuthUserName=$IRC_OMNI_USERNAME \
  --from-literal=AuthPassword=$IRC_OMNI_PASSWORD
```

(Optional) If you want to import the ObjectServer TLS Certificate, use the following command to specify the **tls.crt** file in the secret.

```
PROBE_OMNI_SECRET=noi-probe-secret
oc create secret generic $PROBE_OMNI_SECRET --namespace $PROBE_NAMESPACE \
 --from-literal=AuthUserName=$IRC_OMNI_USERNAME \
 --from-literal=AuthPassword=$IRC_OMNI_PASSWORD \
 --from-file=tls.crt=tls.crt
```

3. Create a secret for the probe to use for basic authentication.

```
PROBE_AUTH_SECRET=sevone-probe-client-basic-auth
oc create secret generic --namespace $PROBE_NAMESPACE $PROBE_AUTH_SECRET --from-
literal=serverBasicAuthenticationUsername=<username>  --from-
literal=serverBasicAuthenticationPassword=<password>
```

where **<username>** is a user name and **<password>** is the password for SevOne NMS to use as basic authentication.

4. Create a Network Policy in the IBM Cloud Pak for AIOps namespace.

Note: Review any other Network Policy that may be denying access to the ObjectServer pod and update the policy to allow ingress connection to the pods.

```
cat << EOF | tee >(oc apply -f -) | cat
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: probe-allow-objectserver
  namespace: ${IRC_NAMESPACE}
spec:
  ingress:
  - from:
    - namespaceSelector:
        matchLabels:
          kubernetes.io/metadata.name: noi-integrations
    ports:
      - protocol: TCP
        port: 4100
  podSelector:
    matchLabels:
      app.kubernetes.io/instance: ${IRC_INSTANCE}
      app.kubernetes.io/managed-by: ir-core-operator
      app.kubernetes.io/name: IssueResolutionCore
  policyTypes:
  - Ingress
EOF
```

5. Create a Probe for SevOne Integrations with the **WebhookProbe** custom resource by running the following commands.

```
PROBE_SEVONE_INSTANCE=sevone-probe

cat << EOF | tee >(oc apply -f -) | cat
apiVersion: probes.integrations.noi.ibm.com/v1
kind: WebhookProbe
metadata:
  name: ${PROBE_SEVONE_INSTANCE}
  labels:
    app.kubernetes.io/name: ${PROBE_SEVONE_INSTANCE}
    app.kubernetes.io/managed-by: netcool-integrations-operator
    app.kubernetes.io/instance: ${PROBE_SEVONE_INSTANCE}
  namespace: ${PROBE_NAMESPACE}
spec:
  helmValues:
```

```
    netcool:
      backupHost: '${IRC_BACKUP_OBJECTSERVER_SVC}.${IRC_NAMESPACE}.svc'
      backupPort: ${IRC_BACKUP_OBJECTSERVER_PORT}
      backupServer: 'AGGB'
      connectionMode: SSLAndAuth
      primaryHost: '${IRC_PRIMARY_OBJECTSERVER_SVC}.${IRC_NAMESPACE}.svc'
      primaryPort: ${IRC_PRIMARY_OBJECTSERVER_PORT}
      primaryServer: 'AGGP'
      secretName: '${PROBE_OMNI_SECRET}'
    probe:
      jsonParserConfig:
        notification:
          jsonNestedHeader: ''
          jsonNestedPayload: ''
          messageDepth: 3
          messageHeader: ''
          messagePayload: json
      integration: sevone
      enableTransportDebugLog: false
      messageLevel: debug
    ingress:
      enabled: true
      host: ''
    arch: amd64
    webhook:
      uri: /probe/sevone
      serverBasicAuthenticationCredentialsSecretName: '${PROBE_AUTH_SECRET}'
      tls:
        enabled: true
        secretName: ''
  license:
    accept: true
  version: 3.1.0
EOF
```

6. Verify that the probe pod is running.

```
oc get pods -l app.kubernetes.io/instance=$PROBE_SEVONE_INSTANCE
```

Obtain the Probe Webhook URL

Use the following command to get the probe webhook URL, which is used to configure a Webhook destination in IBM SevOne NMS.

```
PROBE_HOSTNAME=$(oc get route $PROBE_SEVONE_INSTANCE-mb-webhook -o jsonpath='{.spec.host}')
PROBE_URI=$(oc get route $PROBE_SEVONE_INSTANCE-mb-webhook -o jsonpath='{.spec.path}')
PROBE_WEBHOOK_URL=https://$PROBE_HOSTNAME$PROBE_URI
echo "$PROBE_WEBHOOK_URL"
```

Configure IBM SevOne Network Management System (NMS) to forward events to the Probe for SevOne

IBM SevOne NMS must be configured with a new webhook configuration to send alerts to the Probe for SevOne NMS.

Note: Instructions to configure the webhook with SevOne NMS can differ depending on your SevOne NMS version. For more information about using the SevOne Webhook Definition Manager and configuring alerts in SevOne NMS, see NMS 6.6 Guides and the SevOne NMS User Guide.

Note IBM SevOne requires the webhook destination to use a TLS certificate signed by a trusted Certificate Authority (CA).

1. Log in to the IBM SevOne console.
2. On the top menu, go to **Events > Configuration > Webhook Definition Manager**.
3. On the Policy Browser page, click **Configure Webhook Destination** on the Policies panel.
4. Complete the following steps from the Add Webhook Destination panel:

1. Set a Webhook Destination Name, for example: IBM Cloud Pak for AIOps Probe for SevOne
2. Set a Description, for example: IBM Cloud Pak for AIOps Probe for SevOne NMS
3. Set the URL with the `$PROBE_WEBHOOK_URL` value from the previous Obtaining the Probe Webhook URL section.
4. Select the **Allow insecure webhook connection** option if the probe uses a self-signed certificate.
5. Expand the Headers section and select **Use Basic Auth**. Then, set the **Username** and **Password** fields with the credentials set in the `PROBE_AUTH_SECRET` from the Configure and install probe for SevOne integration section.
6. Click **Test Definition** to send a test HTTP event to the probe. Test both the triggering condition and clear condition webhooks to ensure that the recieved response code is `200` or `204`.
   - A `204` status code indicates that no payload is included in the HTTP response, but the connection is successful.
   - If a response code of `400 Bad request` is received, check the JSON template that is configured. Ensure that the JSON does not contain any syntax errors.
7. Click **Save**.

5. Configure or create policies to send events to the probe by selecting the probe as the webhook destination. For each of the policies in the **Webhooks** section, select the webhook that was created in the previous step. Complete the selection for the **Trigger Conditions** and **Clear Conditions** tabs and select the **One Webhook per Alert** option.

Probe for IBM Turbonomic integration

You can configure the Probe integration for IBM Turbonomic for connecting to IBM Cloud Pak for AIOps.

Note: Since the release of IBM Cloud Pak AIOps 4.1, you can now use the Generic Webhoo Connector to integrate IBM Cloud Pak for AIOps with IBM Turbonomic to digest events. Using the Webhook Connector is preferred over installing a probe to connect with IBM Turbonomic as it simplifies the setup process. For details see Creating Generic Webhook connections.

The Probe integration for IBM Turbonomic is provided by the IBM Netcool Operations Insight Event Integrations Operator.

- Prerequisites
- Common pre-installation tasks
- Configure and install probe For Turbonomic integration
- Obtain the probe Webhook URL
- Configure IBM Turbonomic to forward events to the probe for Turbonomic

Prerequisites

The following systems are prerequisites for this integration.

1. IBM Cloud Pak for AIOps in Openshift Container Platform
2. IBM Turbonomic (IBM Turbonomic 8.6)

Common pre-installation tasks

Follow the instructions in Installing the IBM Netcool Event Integrations Operator and common probe configuration steps to install the operator and gather the ObjectServer connection information before proceeding with the following sections.

Configure and install probe For Turbonomic integration

Note: Using a generic webhook connection is preferred over installing a probe to connect with IBM Turbonomic, so you should consider migrating to this method. See Creating Generic Webhook connections.

Follow the instructions below to create the required secrets, configure the WebhookProbe custom resource and install the probe in the namespace.

1. Create a secret with the ObjectServer credentials for the probe to authenticate with the ObjectServer.

```
PROBE_OMNI_SECRET=noi-probe-secret
oc create secret generic $PROBE_OMNI_SECRET \
 --from-literal=AuthUserName=$IRC_OMNI_USERNAME \
 --from-literal=AuthPassword=$IRC_OMNI_PASSWORD
```

   If you want to import the ObjectServer TLS Certificate (optional), use the following command to specify the **tls.crt** file in the secret.

```
PROBE_OMNI_SECRET=noi-probe-secret
oc create secret generic $PROBE_OMNI_SECRET \
 --from-literal=AuthUserName=$IRC_OMNI_USERNAME \
 --from-literal=AuthPassword=$IRC_OMNI_PASSWORD \
 --from-file=tls.crt=tls.crt
```

2. Create a secret for the probe to use for basic authentication.

```
PROBE_AUTH_SECRET=turbonomic-probe-client-basic-auth
oc create secret generic $PROBE_AUTH_SECRET --from-
literal=serverBasicAuthenticationUsername=<username>  --from-
literal=serverBasicAuthenticationPassword=<password>
```

   where **<username>** is a user name and **<password>** is the password for Turbonomic to use as basic authentication.

3. Create a Network Policy in the IBM Cloud Pak for AIOps namespace.

   Note: Review any other network policy that may be denying access to the ObjectServer pod(s) and update the policy to allow ingress connection to the pods.

```
cat << EOF | tee >(oc apply -f -) | cat
apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
  name: probe-allow-objectserver
  namespace: ${IRC_NAMESPACE}
spec:
  ingress:
  - from:
    - namespaceSelector:
        matchLabels:
          kubernetes.io/metadata.name: noi-integrations
    ports:
      - protocol: TCP
        port: 4100
  podSelector:
    matchLabels:
      app.kubernetes.io/instance: ${IRC_INSTANCE}
      app.kubernetes.io/managed-by: ir-core-operator
      app.kubernetes.io/name: IssueResolutionCore
  policyTypes:
  - Ingress
EOF
```

4. Create a probe for Turbonomic integrations with the **WebhookProbe** custom resource by running the following commands:

```
PROBE_TURBONOMIC_INSTANCE=turbonomic-probe

cat << EOF | tee >(oc apply -f -) | cat
apiVersion: probes.integrations.noi.ibm.com/v1
kind: WebhookProbe
metadata:
  name: ${PROBE_TURBONOMIC_INSTANCE}
  labels:
    app.kubernetes.io/name: ${PROBE_TURBONOMIC_INSTANCE}
    app.kubernetes.io/managed-by: netcool-integrations-operator
    app.kubernetes.io/instance: ${PROBE_TURBONOMIC_INSTANCE}
  namespace: ${NAMESPACE}
spec:
  helmValues:
    netcool:
```

```
      backupHost: '${IRC_BACKUP_OBJECTSERVER_SVC}.${IRC_NAMESPACE}.svc'
      backupPort: ${IRC_BACKUP_OBJECTSERVER_PORT}
      backupServer: 'AGGB'
      connectionMode: SSLAndAuth
      primaryHost: '${IRC_PRIMARY_OBJECTSERVER_SVC}.${IRC_NAMESPACE}.svc'
      primaryPort: ${IRC_PRIMARY_OBJECTSERVER_PORT}
      primaryServer: 'AGGP'
      secretName: '${PROBE_OMNI_SECRET}'
    probe:
      jsonParserConfig:
        notification:
          jsonNestedHeader: ''
          jsonNestedPayload: ''
          messageDepth: 3
          messageHeader: ''
          messagePayload: json
      integration: Turbonomic
      enableTransportDebugLog: false
      messageLevel: debug
    ingress:
      enabled: true
      host: ''
    arch: amd64
    webhook:
      uri: /probe/turbonomic
      serverBasicAuthenticationCredentialsSecretName: '${PROBE_AUTH_SECRET}'
      tls:
        enabled: true
        secretName: ''
  license:
    accept: true
  version: 3.1.0
EOF
```

5. Verify that the probe pod is running.

```
oc get pods -l app.kubernetes.io/instance=$PROBE_TURBONOMIC_INSTANCE
```

Obtain the probe Webhook URL

Use the following command to get the probe webhook URL, which is used to configure a Webhook destination in IBM Turbonomic.

```
PROBE_HOSTNAME=$(oc get route $PROBE_TURBONOMIC_INSTANCE-mb-webhook -o
jsonpath='{.spec.host}')
PROBE_URI=$(oc get route $PROBE_TURBONOMIC_INSTANCE-mb-webhook -o jsonpath='{.spec.path}')
PROBE_WEBHOOK_URL=https://$PROBE_HOSTNAME$PROBE_URI
echo "$PROBE_WEBHOOK_URL"
```

Configure IBM Turbonomic to forward events to the probe for Turbonomic

IBM Turbonomic must be configured with a new webhook configuration to send alerts to the probe for Turbonomic.

Creating a workflow

1. Set the following variables:

```
TURBONOMIC_HOSTNAME=<api endpoint hostname>
TURBONOMIC_ADMIN_USER=<turbonomic username>
TURBONOMIC_ADMIN_PASSWORD=<turbonomic password>
```

```
BASIC_AUTH_USERNAME=<probe basic authentication username>
BASIC_AUTH_PASSWORD=<probe basic authentication password>
```

2. Get the JSESSIONID token

```
JSESSIONID=$(curl \
--silent \
--cookie-jar - \
--insecure \
https://${TURBONOMIC_HOSTNAME}/api/v3/login?hateoas=true \
--data "username=${TURBONOMIC_ADMIN_USER}&password=${TURBONOMIC_ADMIN_PASSWORD}" \
| awk '/JSESSIONID/{print $7}')
```

3. Create a workflow

```
curl \
"https://${TURBONOMIC_HOSTNAME}/api/v3/workflows" \
--insecure \
--compressed \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--header "cookie: JSESSIONID=${JSESSIONID}" \
--request POST \
-vv \
--data @- <<EOF
{
"displayName": "Event Probe",
"className": "Workflow,
"description": "Event webhook probe",
"discoveredBy": {
    "readonly": false
},
"type": "WEBHOOK",
"typeSpecificDetails": {
  "url": "${PROBE_WEBHOOK_URL}",
  "method": "POST",
  "authenticationMethod": "BASIC",
  "username": "${BASIC_AUTH_USERNAME}",
  "password": "${BASIC_AUTH_PASSWORD}",
  "template": "\$converter.toJson(\$action)",
  "trustSelfSignedCertificates": true,
  "headers": [
     {
      "name": "Content-type",
      "value": "application/json"
     }
   ],
  "type": "WebhookApiDTO"
  }
}
EOF
```

4. Get the workflow UID for later use.

```
WORKFLOW_UUID=$(curl \
"https://${TURBONOMIC_HOSTNAME}/api/v3/workflows" \
--insecure \
--compressed \
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--header "cookie: JSESSIONID=${JSESSIONID}" \
--request GET | jq -c ".[] | select(.typeSpecificDetails .
url | contains(\"${PROBE_WEBHOOK_URL}\")) | .uuid" | tr -d '"')
```

Create a policy

1. Log in to the Turbonomic console.

2. Navigate to the *Settings* page and then choose **Policies**.

3. Click **New Automation Policy** and then select the policy type, **Virtual Machine** in this case. This sets the type of entity that your policy will affect. Refer to Working with policies for more information.

   Name the policy

   - **Add Scope** The scope determines which entities this policy will affect. Choose one or more groups, or create new groups and add them to the policy scope. These groups match the type of entity you have set for the policy.
   - **Policy Schedule** The *Select Schedule* fly-out lists all the schedules that are currently defined for your instance of Turbonomic. Expand a schedule entry to see further details, or skip it if you don't need the schedule.
   - **Automation and Orchestration** You can define automation and orchestration settings for different action types within the same policy.
     - **Action Type** See a list of actions that are viable for the policy, and then make your selections.
     - **Action Generation and Acceptance**
       - Do not generate actions as Turbonomic doesn't considers your selected actions in its calculations. For example, if you do not want to generate resize actions for VMs in the policy, analysis will still drive toward the desired state, but will do so without considering resizes.
       - Generate Actions Turbonomic generates your selected actions to address or prevent problems. Choose from the following *Action Acceptance* modes to indicate how you would like the actions to run:
         - **Recommend** — Recommend the action so a user can run it via the given hypervisor or by other means
         - **Manual** — Recommend the action, and provide the option to run that action through the Turbonomic user interface
         - **Automatic** — Run the action automatically For automated resize or move actions on the same entity, Turbonomic waits five minutes between each action to avoid failures associated with trying to run all actions at once. Any action awaiting execution stays in queue.
     - **Before Execution and Action Execution:** By default, generated actions run without the need for orchestration. Turbonomic gives you the ability to set up orchestration to affect the execution of actions.
     - **After Execution**: The default is to 'do nothing'. Other options include:
       - **Create Record in Orchestration** Turbonomic registers the action in the Orchestration log, showing that the given action has been run. For example: A final ActionID could be created and with the final ActionState, the Type and Severity would be updated accordingly in waiops. Those actions would also be updated to Waiops, and it would handle with the alerts with updated state: closed or cleared or keep open.
       - **Use Action Script** Run an action script that is set up for the POST entrypoint. The action script name must match the entity type and action type.
4. Save the policy.
5. Action events will be sent to the probe when an action is run.

Triggering a test event

To trigger a test event to verify that the webhook integration is successful:

1. Get a list of actions by market to test out the workflow. Select an action from the list and copy the UUID, which will be used to trigger an action event manually using an API.

```
curl \
"https://${TURBONOMIC_HOSTNAME}/api/v3/markets/Market/actions?ascending=true" \
--insecure \
--compressed \
```

```
--header 'Accept: application/json' \
--header 'Content-Type: application/json' \
--header "cookie: JSESSIONID=${JSESSIONID}" \
--request GET
```

2. To test triggering an action event based on the UUID in the workflow, use one of the action UUIDs from the previous command (in the actionId attribute value) as in the following command:

```
curl --location --request POST
"https://${TURBONOMIC_HOSTNAME}/api/v3/workflows/${WORKFLOW_UUID}" \
--header 'Content-Type: application/json' \
--header "Cookie: JSESSIONID=$JSESSIONID" \
--insecure \
--data-raw '{
    "operation": "TEST",
    "actionId": "637553640602793"
}'
```

Datadog integration probe

An IBM Tivoli Netcool/OMNIbus probe for integrating with Datadog is available. This probe integration sends HTTP GET requests to Datadog to retrieve events based on the time frame set by start and end mandatory query parameters in the URI.

BETA *Datadog integration probe* is available only for technology preview, and is not for production usage.

Prerequisites

The following systems are prerequisites for this integration.

- IBM Cloud Pak for AIOps version 4.2.0
- Datadog: Cloud Monitoring as a Service

Note: This probe only supports the **`https://api.datadoghq.com/api/v1/events`** v1 API to get events from Datadog.

Note: The IBM Netcool Operations Insight Event Integrations Operator must be deployed before deploying this integration. See Installing the IBM Netcool Operations Insight Event Integrations Operator

To deploy a new instance of the Generic Webhook Probe, review the custom resource YAML template supplied with the operator package, update the parameters if required, and save the configurable parameters in a file, for example **`probes_v1_webhookprobe_datadog_integration.yaml`**.

Then deploy the custom resource using the updated file:

```
kubectl apply -f deploy/crs/probes_v1_webhookprobe_datadog_integration.yaml --namespace
<NAMESPACE>
```

Note: If you are installing using the CLI, you can get the custom resource Yaml file from the **`ibm-netcool-integrations/inventory/netcoolIntegrationsOperatorSetup/files/op-cli/deploy/crs`** directory in our CASE archive. If you are installing using the Operator Lifecycle Manager (OLM) Web console, the custom resource is provided in the YAML view.

Pre-installation tasks

The Probe for SevOne, Probe for Turbonomics, and Probe for Datadog have the same pre-installation requirements. See Probe Common Tasks before proceeding with the following sections.

Configuring and installing the Probe for Datadog integration

Follow the instructions below to create the required secrets, configure the **WebhookProbe** custom resource and install the probe in the namespace.

1. Create a secret with the ObjectServer credentials for the probe to authenticate with the ObjectServer.

```
PROBE_OMNI_SECRET=noi-probe-secret
kubectl create secret generic $PROBE_OMNI_SECRET --from-
literal=AuthUserName=$NOI_OMNI_USERNAME --from-literal=AuthPassword=$NOI_OMNI_PASSWORD --
from-file=tls.crt=tls.crt
```

2. Create a secret for the Datadog certificate.

```
echo -n | openssl s_client -connect api.datadoghq.com:443 | sed -ne '/-BEGIN
CERTIFICATE-/,/-END CERTIFICATE-/p'  > datadog.pem
PROBE_CLIENT_TLS_SECRET=datadog-probe-client-tls-cert
kubectl create secret generic $PROBE_CLIENT_TLS_SECRET --from-file=server.crt=datadog.pem
```

3. Create a secret for the transport properties.

```
TRANSPORT_PROPS=restMultiChannelHttpTransport.json
cat << EOF | tee > $TRANSPORT_PROPS
{
  "GLOBAL":
  {
    "httpVersion":"1.1",
    "httpHeaders":"",
    "responseTimeout":"60",
    "securityProtocol":"TLSv1.2",
    "keepTokens":"",
    "autoReconnect":"OFF",
    "gatherSubsTopicInfo":"false"
  },
  "RESYNC":
  {
      "RESYNC_CHILD_BY_FDN":
      {
          "uri":"https://api.datadoghq.com:443/api/v1/events?start=++date_happened:
<initial-start-epoch-time>++&end=++now++&page=0",
          "method":"GET",
          "headers":"Accept=application/json,DD-API-KEY=<datadog-api-key>,DD-APPLICATION-
KEY=<datadog-application-key>",
          "content":"{}",
          "contentFile" : "",
          "interval":"60",
          "attempts":"0",
          "requireSSL":"true",
          "asEventStream" : "false",
          "enablePagination" : "true"
      }
  }
}
EOF

PROBE_HTTP_REQUEST=http-requests
kubectl create secret generic $PROBE_HTTP_REQUEST --from-
file=httpRequests=$TRANSPORT_PROPS
```

   Where:

   - **++date_happened:<initial-start-epoch-time>++** allows the probe to assign the value of RecordData.date_happened in the probe data backup file to the start query parameter to continue from previous resynchronization if RecordData.date_happened is available. Otherwise, the probe assigns **<initial-start-epoch-time>** to the start query parameter.
   - **<datadog-api-key>** is the Datadog API key.
   - **<datadog-application-key>** is Datadog Application key.

   Note: The end query parameter can be set to an epoch time or ++now++ which represents the current epoch time.

4. Create a Network Policy in the IBM Tivoli Netcool/OMNIbus namespace to allow the probe to access the TLS Proxy service and port in IBM Tivoli Netcool/OMNIbus namespace. Note: Review any other Network Policy that may be denying access to the ObjectServer TLS Proxy pod and update the policy to allow ingress connection to the TLS Proxy pod.

```
cat << EOF | tee >(kubectl apply -f -) | cat
kind: NetworkPolicy
apiVersion: networking.k8s.io/v1
metadata:
  name: noi-allow-proxy
  namespace: ${NOI_NAMESPACE}
spec:
  podSelector:
    matchLabels:
      app.kubernetes.io/instance: ${NOI_INSTANCE}
      app.kubernetes.io/name: proxy
  ingress:
    - ports:
        - protocol: TCP
          port: ${NOI_PROXY_PRIMARY_PORT}
        - protocol: TCP
          port: ${NOI_PROXY_BACKUP_PORT}
  policyTypes:
    - Ingress
EOF
```

5. Create a Probe for Datadog Integrations with the WebhookProbe custom resource by running the following commands.

```
PROBE_DATADOG_INSTANCE=datadog-probe

cat << EOF | tee >(kubectl apply -f -) | cat
kind: WebhookProbe
apiVersion: probes.integrations.noi.ibm.com/v1
metadata:
  name: webhookprobe
  labels:
    app.kubernetes.io/name: ${PROBE_DATADOG_INSTANCE}
    app.kubernetes.io/managed-by: netcool-integrations-operator
    app.kubernetes.io/instance: ${PROBE_DATADOG_INSTANCE}
  namespace: ${NAMESPACE}
spec:
  license:
    accept: true
  version: 3.3.0
  helmValues:
    global:
      image:
        secretName: ''
      serviceAccountName: ''
      enableNetworkPolicy: true
    image:
      useDefaultOperandImages: true
      probe: >-
        cp.icr.io/cp/noi-int/netcool-probe-
messagebus@sha256:745a7dae41f1a0c8f1517a9dc2368d227e39be4798b74175f903bc5ac940b95b
      utility: >-
        cp.icr.io/cp/noi-int/netcool-integration-
util@sha256:6e0b9883a34810c468b5386b4b6d9ed216f68df015f6622b83656d400d01dcd2
      pullPolicy: IfNotPresent
    netcool:
      connectionMode: SSLAndAuth
      primaryServer: 'AGGP'
      primaryHost: '${NOI_PROXY_SVC}.${NOI_NAMESPACE}.svc'
      primaryPort: 4100
      backupServer: 'AGGB'
      backupHost: '${NOI_PROXY_SVC}.${NOI_NAMESPACE}.svc'
      backupPort: 4100
      secretName: '${PROBE_OMNI_SECRET}'
    probe:
      integration: datadog
      messageLevel: warn
      setUIDandGID: false
```

```
      sslServerCommonName: ''
      locale: en_US.utf8
      enableTransportDebugLog: false
      heartbeatInterval: 10
      recordData: 'date_happened'
      configPVC:
        name: ''
        rulesFile: ''
        parserConfigFile: ''
        dataBackupFile: ''
      rulesConfigmap: ''
      jsonParserConfig:
        notification:
          messagePayload: json
          messageHeader: ''
          jsonNestedPayload: ''
          jsonNestedHeader: ''
          messageDepth: 3
        resync:
          messagePayload: 'json.events'
          messageHeader: ''
          jsonNestedPayload: ''
          jsonNestedHeader: ''
          messageDepth: 3
      httpClientCredentialsSecretName: ''
      initialResync: true
      resyncInterval: 120
      poddisruptionbudget:
        enabled: false
        minAvailable: 1
      selfMonitoring:
        discardHeartbeatEvent: false
        populateMasterProbeStat: false
        logProbeStat: false
        generateThresholdEvents: false
      configs: 'ProbeWatchHeartbeatInterval:0 EventLoadProfiling:''true'''
    ingress:
      enabled: false
      host: ''
    webhook:
      uri: /probe
      httpVersion: '1.1'
      respondWithContent: 'OFF'
      validateBodySyntax: 'ON'
      validateRequestURI: 'ON'
      idleTimeout: 180
      tls:
        enabled: true
        secretName: ''
      serverBasicAuthenticationCredentialsSecretName: ''
      replicaCount: 1
      autoscaling:
        enabled: false
        minReplicas: 1
        maxReplicas: 3
        cpuUtil: 60
    httpClient:
      enabled: true
      host: ''
      port: 80
      sslSecretName: '$PROBE_CLIENT_TLS_SECRET'
      httpRequestsSecretName: '$PROBE_HTTP_REQUEST'
    resources:
      requests:
        cpu: 100m
        memory: 128Mi
        ephemeral-storage: 100Mi
      limits:
        cpu: 200m
        memory: 256Mi
        ephemeral-storage: 200Mi
```

```
    arch: amd64
EOF
```

6. Verify that the probe pod is running.

```
kubectl get pods -l app.kubernetes.io/instance=$PROBE_DATADOG_INSTANCE
```

7. If you want to store the probe data backup file in Persistent Volume Claim (PVC), follow the instructions in Providing custom Message Bus Probe configuration files in a persistent volume. Then set configPVC.name to a pre-created PVC name and configPVC.dataBackupFile to the data backup filename.

Installing the IBM Netcool Event Integrations Operator and common probe configuration steps

This topic describes how to install the IBM Netcool Event Integrations Operator which contains the probes you will need to connect to, and acquire event data from, event sources. It also describes how to gather the IBM Cloud Pak for AIOps ObjectServer connection information that the individual probe installations require.

- 1. Install the IBM Netcool Event Integrations Operator
- 2. Gather IBM Cloud Pak for AIOps ObjectServer connection information

1. Install the IBM Netcool Event Integrations Operator

Install the IBM® Netcool® Operations Insight® Event Integrations Operator.

- For more information, see Installing the IBM Netcool Operations Insight Event Integrations Operator but skip the Install Netcool Operations Insight step.
- Or see Installing the IBM Netcool Operations Insight Event Integrations Operator to a private container registry.

2. Gather IBM Cloud Pak for AIOps ObjectServer connection information

Before you configure the probe and connect to the ObjectServer, you need to gather the following information:

- Primary and backup ObjectServer service name and port
- ObjectServer client credentials
- TLS certificate used by the ObjectServer

In the IBM Cloud Pak for AIOps, the ObjectServer is deployed by the issue resolution core component. Complete the following instructions to get the ObjectServer connection details.

1. Set the IBM Cloud Pak for AIOps namespace variable to the namespace where IBM Cloud Pak for AIOps is installed.

```
IRC_NAMESPACE=<namespace>
```

2. Determine the issue resolution core instance name.

```
IRC_INSTANCE=$(oc get issueresolutioncore -n $IRC_NAMESPACE -o custom-
columns=name:metadata.name --no-headers)
```

3. Run the following commands to gather service information and credentials.

```
 IRC_PRIMARY_OBJECTSERVER_SVC=$IRC_INSTANCE-ir-core-ncoprimary
 IRC_BACKUP_OBJECTSERVER_SVC=$IRC_INSTANCE-ir-core-ncobackup
 IRC_PRIMARY_OBJECTSERVER_PORT=$(oc get svc -n $IRC_NAMESPACE
$IRC_PRIMARY_OBJECTSERVER_SVC -o jsonpath='{.spec.ports[?(@.name=="primary-tds-
port")].port}')
 IRC_BACKUP_OBJECTSERVER_PORT=$(oc get svc -n $IRC_NAMESPACE $IRC_BACKUP_OBJECTSERVER_SVC
-o jsonpath='{.spec.ports[?(@.name=="backup-tds-port")].port}')
 IRC_OMNI_USERNAME=aiopsprobe
```

```
 IRC_OMNI_PASSWORD=$(oc get secret -n $IRC_NAMESPACE $IRC_INSTANCE-ir-core-omni-secret -o
jsonpath='{.data.OMNIBUS_PROBE_PASSWORD}' | base64 --decode && echo)
```

4. Extract the ObjectServer TLS certificate from the secret into your current directory. This file is used when creating a secret for the probe to connect with the ObjectServer in the following sections.

```
oc extract secret/$IRC_INSTANCE-ir-core-ncoprimary-tls -n $IRC_NAMESPACE --to=. --
keys=tls.crt
```

5. Proceed to install the probe.

Managing AIOps insights

IBM Cloud Pak® for AIOps includes an *AIOps insights* dashboard in the IBM Cloud Pak Automation console. This dashboard provides a set of predefined interactive charts that show you, at a glance, what is going on in your environment. For more information about using the dashboard, see AIOps insights dashboard.

Note: The **AIOps insights** dashboard is not related to the IBM AIOps Insights offering. If you are looking for information about that offering, see the documentation for IBM AIOps Insights.

Enabling AIOps insights

During installation of IBM Cloud Pak® for AIOps, **AIOps insights** is enabled by default and is accessible to all users.

Disabling AIOps insights

Access to **AIOps insights** can be restricted by changing the permissions of a user. For more information, see Roles and permissions.

Scopes and filtering

Once you enable data to flow into the **AIOps insights** dashboard, you can refine your insights by using filters and scopes to segment that data. For more information, see AIOps insights Scopes and Filter.

AIOps insights dashboard

Use the **AIOps insights** dashboard to view key insights provided by IBM Cloud Pak® for AIOps. View insights for a 24-hour, 7-day, or 30-day period. This information can show the value of using IBM Cloud Pak for AIOps for your IT operations.

Note: The **AIOps insights** dashboard is not related to the IBM AIOps Insights offering. If you are looking for information about that offering, see the documentation for IBM AIOps Insights.

The dashboard can be accessed from the administration page of IBM Cloud Pak for AIOps, under **Overview** > **Quick navigation** > **AIOps insights**. Alternatively, click the navigation icon (four horizontal bars) in the upper-left corner of the screen to go to the main navigation menu. Then click **Operate** > **AIOps insights**.

Note: Previously, **AIOps insights** was available only as a technology preview. If you enabled this version, it must be disabled after you upgrade to the latest version of IBM Cloud Pak for AIOps. To disable, follow these instructions in the 3.4 documentation, Disable AIOps insights.



Figure. AIOps Insights dashboard

Prerequisites

All relevant data connections and activity (incident opened, active runbook, and so on) must be operational to allow live data to flow into the chart. By default, all users have access to **AIOps insights**. However, Administrators can restrict access, by changing a user's permissions. For more information, see Roles and permissions.

Note: If you have not yet setup connections or implemented activity (or there is limited information) you can view examples of each **AIOps insights** chart, containing sample data, by taking the **AIOps insights** product tour. Click the signpost icon beside your avatar, then **Product Tours** > **Discover AIOps insights**.

Summary bar

The summary bar of the dashboard displays key indicators and shows how IBM Cloud Pak for AIOps is improving the efficiency of your IT operations teams. It summarizes the charts currently showing on the dashboard. An arrow indicates an increase or decrease in activity.



Figure. Summary bar

For more detailed information on metrics in the summary bar, view the individual charts.

- Mean time to restore
- Incident activity
- Noise reduction
- Runbook automation
- Disruption costs (Enable in **AIOps insights settings** > **Optional Charts**)

Mean time to restore

The *Mean time to restore* chart displays the average time that is needed to isolate, repair, and fully resolve IT incidents.

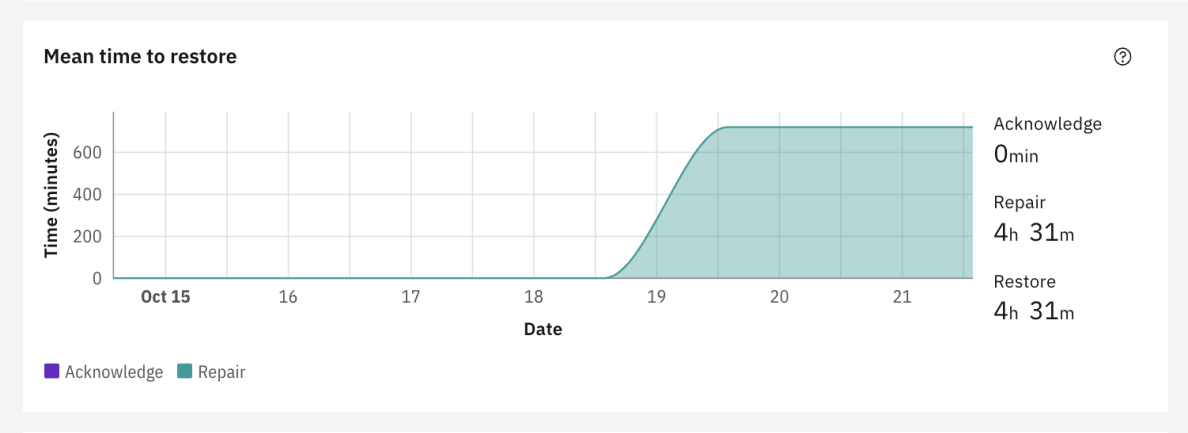


Figure. Mean time to restore chart

Chart definitions:

Acknowledge: the mean time that it takes to begin work on an incident. Toggle the purple *Acknowledge* box in the legend to display only this data on the chart.

Repair: the mean time taken to reestablish a service. Toggle the green *Repair* box in the legend to display only this data on the chart.

Restore: the total mean time (*Acknowledge* time and *Repair* time combined) taken to restore service.

Note: All mean values that are generated are based on dividing the item total by the number of data points on the chart. Because of the time intervals, this is always the selected time period +1 unit. So 24-hour totals are divided by 25, 7-day totals are divided by 8, 30-day totals are divided by 31.

Incident activity

Within IBM Cloud Pak for AIOps, an incident is a collection of alerts, insights, and potential solutions to help drive incident remediation.

By clicking the chart icon in the *Incident activity* chart, you can select to view by *Time series* (default view) or *Summary*.

- The *Time series* chart displays the number of incidents that were opened and closed during the selected time period.

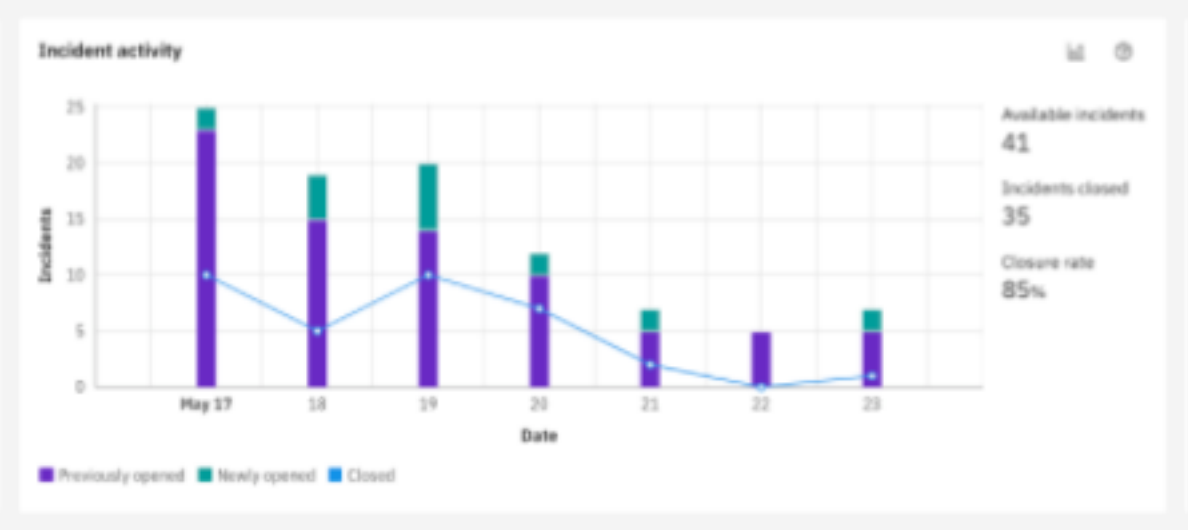


Figure. Incidents closure rate

- The *Summary* chart displays open and closed incidents according to their severity level, over the selected time period.

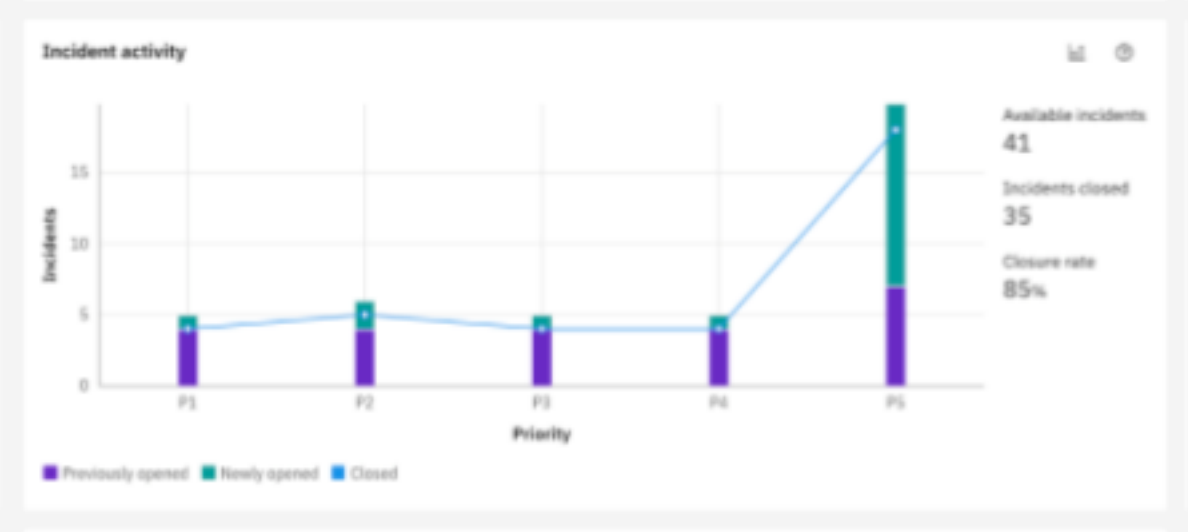


Figure. Summary setting

Chart definitions:

Previously opened: incidents carried over from the preceding time unit (for example, in a 7-day period, it would be incidents that are carried over from the day before), and previously opened incidents that are not yet resolved. Hover the cursor over the purple bar to see specific numbers for each day. Toggle the purple *Previously opened* box in the legend to display only this data on the chart.

Newly opened: incidents opened in the selected time period. Hover the cursor over the green bar to see specific numbers for each day. Toggle the green *Opened* box in the legend to display only this data on the chart.

Closed: the number of incidents that closed in the selected time period. Expressed as a line graph. Toggle the blue *Closed* box in the legend to display only this data on the chart.

Available incidents: the total number of incidents that were in an open state over the selected period. Combines the number of previously opened incidents – carried over when the selected time period began - with the aggregate number of new incidents (combined value of all the green bars) opened.

Incidents closed: the aggregated number of incidents closed in the selected period.

Closure rate: *Incidents closed* expressed as a percentage of the *Total incidents*.

Noise reduction

The *Noise reduction* chart displays how IBM Cloud Pak for AIOps reduces the number of IT events and alerts that your operations staff must evaluate, speeding up recovery time and reducing employee fatigue.

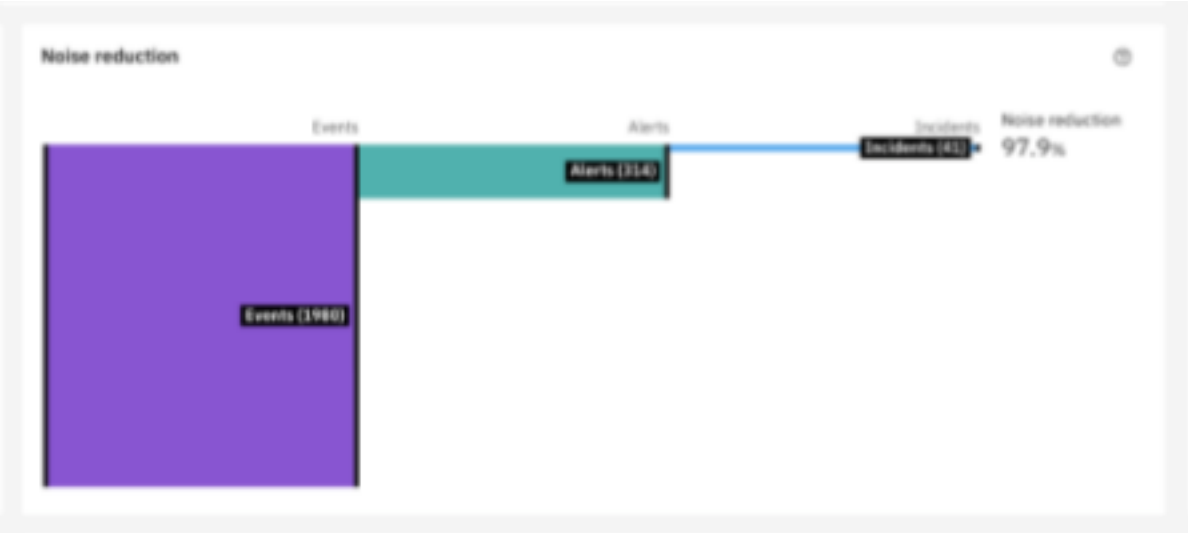


Figure. Noise reduction dashboard

Note: Updated events first created more than 30 days earlier are not incremented into the *Noise reduction* chart. For more information, see this Known issue.

Chart definitions:

Noise reduction occurs in three phases, as depicted in the chart:

1. *Events*: Duplicate or irrelevant events are removed.
2. *Alerts*: Events are correlated and grouped into alerts based on when they occurred and the relationships between the associated IT resources.
3. *Incidents*: Alerts are grouped into prioritized incidents that also include insights and potential solutions to help drive incident remediation.

The *Noise reduction* percentage is the reduction that occurs in the transition from the original number of events that are generated to the final number of incidents.

IBM Cloud Pak for AIOps includes three noise reduction algorithms, two of which require no pre-training. You can also create policies to suppress events and precisely control how alerts are grouped into incidents. For more information, see About event grouping, Deduplication, and About policies.

Incidents and noise reduction:

Leveraging **Incidents** is fundamental to deriving value from IBM Cloud Pak for AIOps. If you limit the number of policies that create incidents, by default you also limit the level of noise reduction that can be tracked and presented in the *Noise Reduction* chart.

Policies connect events and alerts to incidents. Without that connection, the value of the *Noise Reduction* chart is unnecessarily restricted. It can still display the number of *Events* and *Alerts* but not the level of *Noise Reduction*, which will instead display as empty (double dashes).



Figure. Noise reduction dashboard with incidents restricted

Note: Alerts that surface in *Resource Management* may not flow into the Noise Reduction chart. This is because they may not be part of a policy that directly ties them to an incident.

Runbook usage

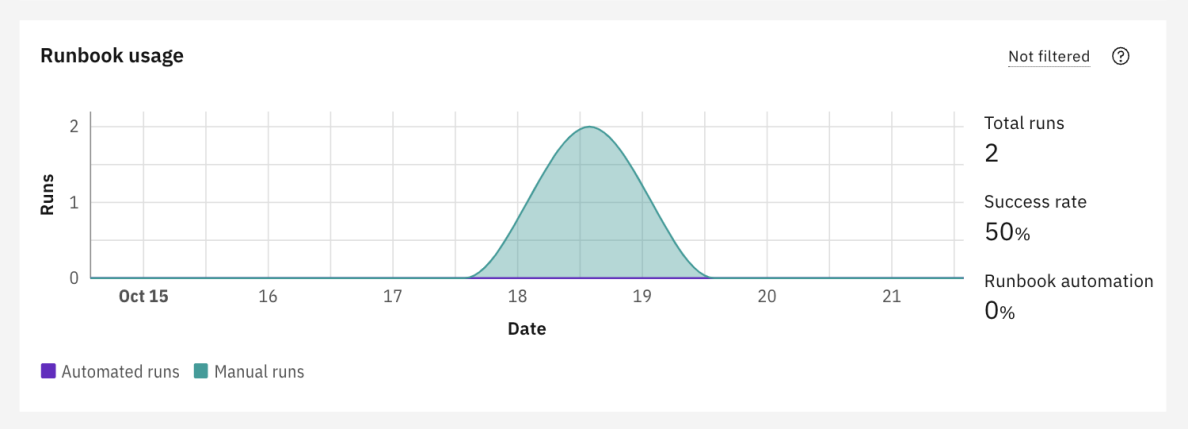


Figure. Runbook usage

Use the runbook usage chart to view the number of automated versus manual runbook runs and the success rates of them.

Chart definitions:

Automated runs: Runbooks that were run without any operator interaction, in the selected period. Toggle the purple *Automated runs* box in the legend to display only this data on the chart.

Manual runs: Runbooks that were implemented manually by an operator who followed an exact procedure, in the selected period. Toggle the green *Manual runs* box in the legend to display only this data on the chart.

Total runs: The combined number of automated and manual runs, in the selected period.

Success rate: The percentage of automated and manual runbooks successfully implemented.

Runbook automation: Percentage of total runbooks that were automated - that needed no operator interaction.

Disruption costs

BETA *Disruption costs* chart is available only for technology preview, and is not for production usage. Go to **AIOps insights settings** > **Optional charts** to turn the chart on or off.

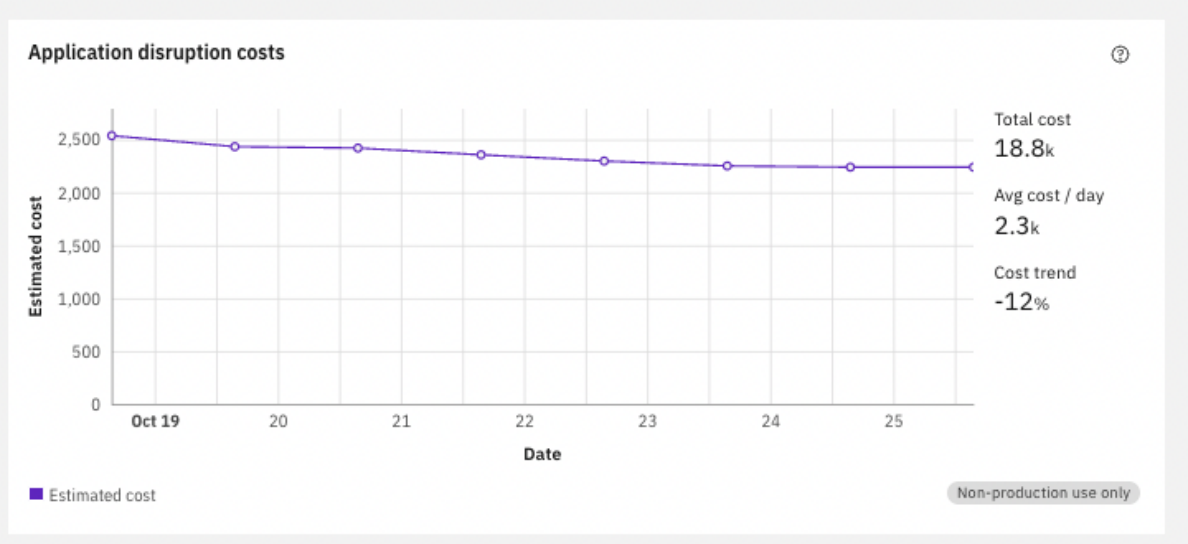


Figure. Disruption costs chart

Note: Default chart title displayed here. Alternatively, *Service disruption costs* may display, depending on set up.

An application is assumed to be disrupted when one or more of its resources are alerting, and one or more of the associated alerts are included in a priority 1 (P1) incident. Disruption costs are calculated based on the amount of time that applications are in this state and the estimated disruption costs for the applications. For example, if one application is in this state for a

total of 10 minutes, and the application includes an estimated disruption cost of 10 units per minute, a total disruption cost of 100 units is recorded.

Important:

- The user must set the estimated disruption cost before using the *Application disruption costs* chart. For more information, see Entering estimated disruption costs for applications.
- If you change the estimated disruption cost for an application, the new value will be applied to previous and future instances of disruption.

  Example:

  If the estimated disruption cost for an application is initially set to **10** units per minute, and the application was disrupted for 5 minutes during a given time period, a total disruption cost of 50 units is recorded for the application (*10 units per minute* multiplied by *5 minutes*).

  If the estimated disruption cost for the application changes, the new value retroactively applies. So, if the estimated disruption cost increases to **20** units per minute, the total disruption cost increases to 100 (*20 units per minute* multiplied by *5 minutes*). Similarly, if the estimated disruption cost reduces to **5** units per minute, the total disruption cost reduces to 25 (*5 units per minute* multiplied by *5 minutes*).

Chart definitions:

Estimated cost: The costs per day based on estimated disruption costs.

Total cost: The cumulative cost of the disruption over the selected period.

Avg cost: The cost per day or hour when the total estimated costs are averaged out over the selected period.

Cost trend: The percentage change in the cost of the disruption from the start of the period to the end of the period.

Chart tooltips

White dots highlight particular points on a chart. Clicking on a chart and then clicking one of these dots generates a tooltip data box for that particular point in time. **Note:** This is not available for Noise reduction chart as it does not use a line graph.

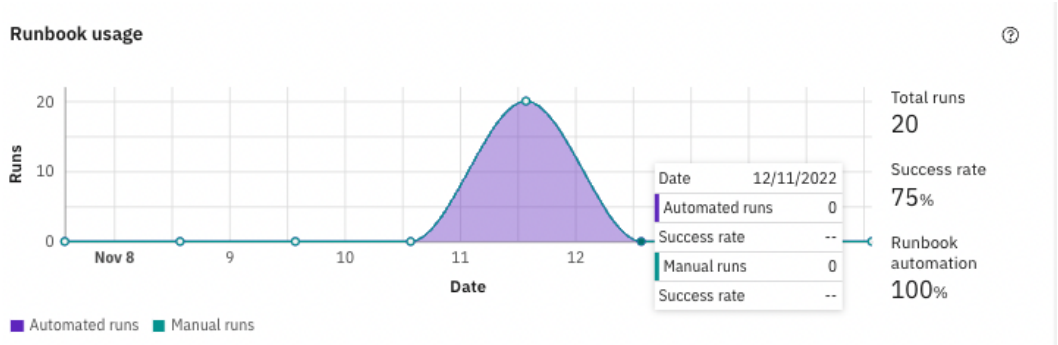


Figure. Tooltip

Tooltip information is also available for metrics in the sidebar of a chart.

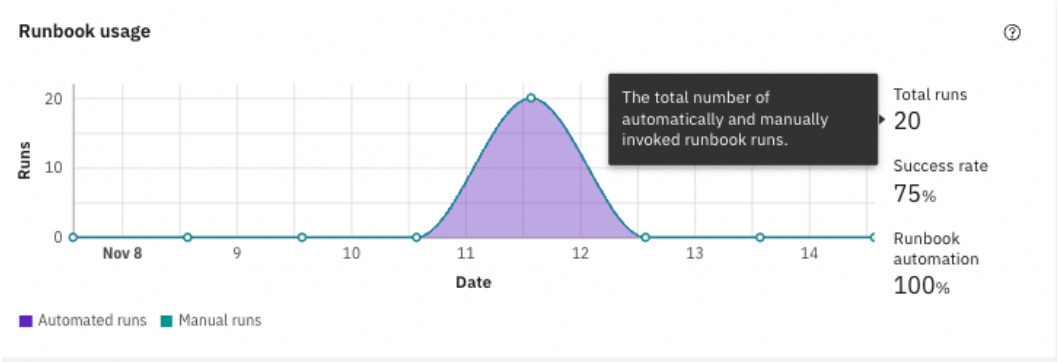


Figure. Tooltip sidebar

AIOps Insights Scopes and Filter

Use customized scopes and the filter bar in **AIOps Insights** to refine your charts, to achieve more focused insights.

Note: The **AIOps Insights** dashboard is not related to the IBM AIOps Insights offering. If you are looking for information about that offering, see the documentation for IBM AIOps Insights.

Create a scope

A scope is a customized view of data. Use a scope to segment data and base your metrics on a meaningful set of IT resources.

Initially, **AIOps Insights** includes a single scope, *All resources*. When selected, metrics are scoped to all of the IT resources monitored by IBM Cloud Pak® for AIOps. Create custom scopes to display metrics for smaller, more meaningful sets of IT resources.

Prerequisites

Creating and managing scopes requires administrative privileges. For more information, see Roles and permissions.

Procedure

Do the following to create a custom scope.

1. On the toolbar, when you first open **AIOps Insights** go to the dropdown menu with the words *All resources*. This indicates that all data that is related to your Cloud Pak for AIOps connection is included in the charts. Click the dropdown menu.

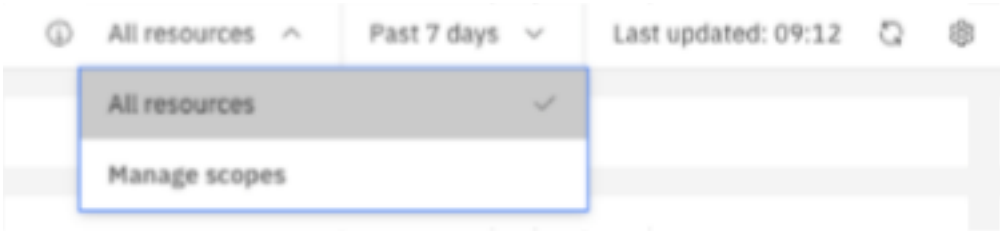


Figure. Scope dropdown

2. From the dropdown menu, select **Manage scopes**. This opens the **Scopes** panel in *AIOps insights settings*, and lists all scopes available to you. **Note**: another route to the Scopes panel is to click *AIOps Insights settings* and then click **Scopes**.

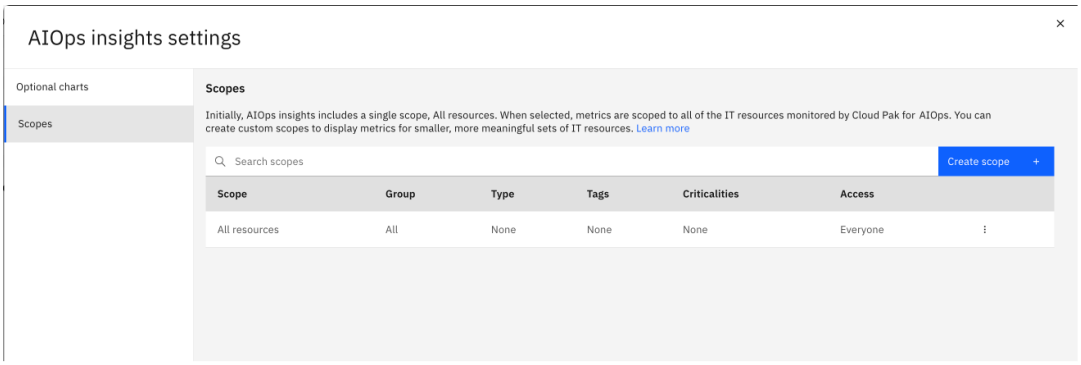


Figure. Scope panel

3. In the **Scopes** panel, click **Create scope**.

4. The *Name and description* panel opens. Enter a short title for your scope – *Banking Sales*, for example – and a description (optional), explaining what this scope shows. When completed, click **Next**.
5. In the **Resources** panel, select either *Applications* or *Resource groups*.
    - Under *Tags*, use the dropdown menu to select how to filter by selected tags:
        - For **Applications**, select either **Include applications with all of the tags that are selected below** or **Include applications with any of the tags that are selected below**.
        - For **Resource groups**, select either **Include resource groups with all of the tags that are selected below** or **Include resource groups with any of the tags that are selected below**.
    - If a tag is attached to an application or resource group, it appears under *Tag*. To find a specific tag, enter the name in the *Search tags* field.
    - Under *Tag*, select an available tag for the scope. A blue border around the tag name indicates it is selected. Click again, to deselect.
    - To scope by *Business criticalities*, click *Criticalities* and select from the criticalities you assigned to define levels of criticality, for example Tier1, Tier2, and so on. For more information on criticalities, see Defining business criticality.
6. Click **Next**.
7. Use the **Access control** panel to control who can access the scope that you are creating. You can also use this panel to enable and disable a scope, which allows you to create and store scopes for future use.

    Once the scope is enabled, use the dropdown menu to select who can access the scope. There are three choices: **Only me**, **Specified user groups** and **Everyone**.

    Use the search box to find any specified user groups who you want to allow access to this scope. Any group that is selected is displayed under *User groups*. A blue border around the user group indicates it is selected. Click again, to deselect.

    When you complete your selections, click **Create scope**.

The scope that you created is now added to the list of scopes in the main **Scopes** panel. Click the three vertical dots to edit or delete a scope. **NOTE**: You can access a description of the scope by hovering your mouse over its title. Where a number appears, hover the cursor over it, to see what this references (a tag, for example).

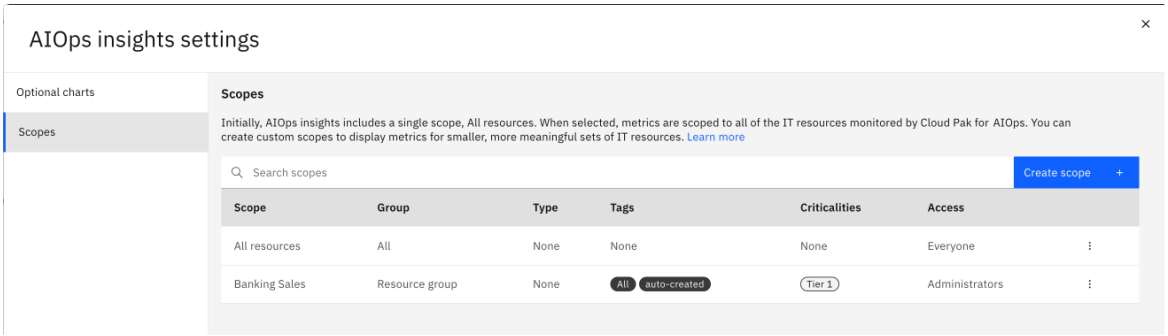


Figure. Scopes panel

On the AIOps Insights dashboard toolbar, the new scope is added to the dropdown menu. Only users who have access to this scope see it in the scopes list.

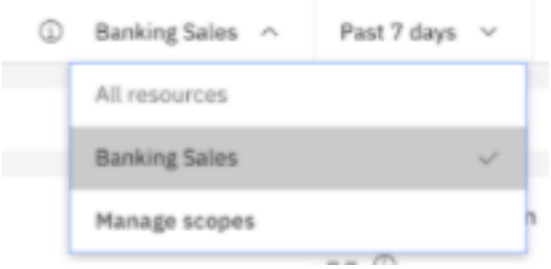


Figure. Scopes dropdown customized

Click the information button beside the scope to view a summary of the key information about the scope.

Adding a filter

Use the filter to refine by incident priority. Changes take effect immediately on the dashboard. **Note**: The filter is not applied to runbook usage metrics.

1. Click the filter icon to access the **Filter** panel.
2. Click *Incident priority* and then select one or more priority (**P**) level. A blue border around the level name indicates it is selected. To clear selections, click the level name again and the border disappears.
3. Click **Apply**.

The incident priority filters that you chose are applied to the dashboards (excluding *Runbook usage*).

Managing AI modelling

Leverage your data to build models that IBM Cloud Pak® for AIOps can use to learn patterns in your data. IBM Cloud Pak for AIOps can then apply these patterns to provide insights to your Operations and SRE teams.

For more information about configuring AI training, see the following topics:

- Preparing the AI environment
- About AI algorithms
- Preparing AI algorithms
- Managing AI algorithms
- Displaying data assets

About AI models

IBM Cloud Pak® for AIOps provides a wide range of AI algorithms to gather data and learn from your environment, and provide insights to support resolution of health issues across your application, service, and network infrastructure.

For each of the following AI algorithms, this topic describes the algorithm, explains how it will help you, and, where configuration is necessary, points to the relevant configuration tasks.

- Change risk
- Event grouping
- Log anomaly detection
- Metric anomaly detecion
- Probable cause
- Similar tickets

Change risk

This AI algorithm provides an assessment of the risk of implementing a proposed change; for example, a code change or a new software version.

Hundreds of changes can affect an application during its lifecycle. Change risk takes historical data about all of those changes and helps you determine how likely it is that a specific change would cause a problem, based on how successfully that similar change was deployed in the past. Using the assessment score provided by this AI model, you can determine how safe it is to proceed with the change.

Training this AI algorithm will help you ensure that risky changes for an application are assessed before deployment. To train this algorithm, you must initially configure it using a wizard, as described in Configuring change risk.

Once this algorithm is successfully trained and the resulting AI model deployed, then when someone opens a change request ticket, a *Change risk assessment* appears in the proactive ChatOps channel. For more details, see Change risk.

Event grouping

Three AI algorithms are provided that group events together and present the groups within a single incident.

The three algorithms are the following:

- Temporal grouping
- Topological grouping
- Scope-based Grouping

Temporal grouping

This AI algorithm groups events that are discovered to co-occur over time. When a problem arises, there are typically multiple parts of a system or environment that are impacted. When events in different areas co-occur, it makes sense to look at them together and treat them as one problem to try to determine what might have happened.

Grouping co-occurring events together reduces the number of tickets and incidents opened and the number of people looking at the same problem, thereby significantly reducing noise in your monitoring systems. It helps you to understand the context of an issue so you can prioritize, triage, and resolve it more quickly.

To train this algorithm, you must initially configure it using a wizard, as described in Configuring temporal grouping.

This algorithm is enabled so when related events are grouped based on when they occur, Site reliability engineers (SREs) and other users responsible for application and service availability will be able to view the details in the event Viewer, as described in About alerts.

Topological grouping

This AI algorithm groups your events based on the resource groups in which events events occur. For example if you have a resource group made up of all the resources within a given Kubernetes namespace, then any events on pods, microservices or other resources in that namespace will be grouped together in a single topological group.

Topological grouping helps you understand when events are connected based on their topology, providing valuable context for why related events might occur together.

This algorithm is enabled so when related events are grouped based on their topology, Site reliability engineers (SREs) and other users responsible for application and service availability will be able to view the details in the Alert Viewer, as described in About alerts.

Scope-based grouping

This AI algorithm automatically groups events relating to an incident if they have the same defined scope and occur during the same period of time. A scope can be used to identify where events originate based on a common attribute, for example, the location of a server room.

By understanding when events are related based on both time and location, you can more quickly diagnose incidents.

This algorithm is enabled so when related events are grouped based on their scope, Site reliability engineers (SREs) and other users responsible for application and service availability will be able to view the details in the Alert Viewer, as described in About alerts.

Log anomaly detection

This pair of AI algorithms gather log data from one or multiple components in the application architecture, identifies a baseline of expected log message types, and uses this baseline to discover abnormal behavior in your live log data. For more information on log anomaly detection algorithms, see About log anomaly detection algorithms.

Metric anomaly detection

Metric anomaly detection first learns normal patterns of metric behavior by analyzing metric values at regular intervals. If that behavior significantly changes, it raises anomalies or alerts.

For more information, see About metric detection algorithms.

Probable cause

This AI algorithm identifies the event with the greatest probability of being the cause of an incident.

Probable cause analyzes event and topology data to understand how events are related to each other. It then uses that understanding to try to determine the root cause of a problem.

All events in an incident are given a probability score that indicates the likelihood of that event being the cause of the incident. events that occur on resource that is lower down in the topology are giving a higher probability score. events are also categorized and are scored based on the category that they fall into. This ensures that if two events occur on the same resource in the topology, the event with the more significant category will have a higher probability score.

Once the SRE has identified the probable cause event, they can take action on the resource that is generating that event and quickly resolve the incident.

The probable cause AI algorithm is enabled by default, so that information about the probable cause of a problem is automatically included in your incident details, which are surfaced in your ChatOps interface, as described in Managing incidents.

Similar tickets

This AI algorithm discovers details about similar messages, anomalies, and events within your tickets that occurred in the past and are impacting the current application.

When an incident occurs, it can be helpful to review details for similar tickets to help determine a resolution. This model aggregates information about similar messages, anomalies, and events for a component or application. This model can also extract the steps used to fix previous incidents, if documented.

Training this AI model will help you discover historical incidents to aid in the remediation of current problems. To train this algorithm, you must initially configure it using a wizard, as described in Setting up training for similar tickets.

Site reliability engineers (SREs) and other users responsible for application and service availability are able to access incidents in their ChatOps interface and within each incident view historical similar tickets, as described in Managing incidents.

Prerequisites for AI modelling

For each different AI algorithm, you must integrate with the right tools and gather specific types of data so the AI can begin learning.

This topic describes the prerequisites for each of the following AI algorithms.

- Change risk and Similar tickets
- Log anomaly detection
- Alert grouping and Probable cause

Change risk and Similar tickets

Both the Change risk algorithm and the Similar ticket algorithm require ticket data from ServiceNow. Before attempting to configure these algorithms, ensure that you have created a functioning data connection to ServiceNow. For more information, see Creating a ServiceNow connection.

Log anomaly detection

The Log anomaly detection algorithms require log data from one or more of your log systems. Before attempting to configure these algorithms, ensure that you have created a functioning data connection to at least one of the log systems listed in the following table.

Log system	For more details, see:
Falcon LogScale	Creating Falcon LogScale connections
LogDNA	Creating LogDNA connections
Splunk	Creating Splunk connections

You can also load log data into the system using one of these generic data loading methods.

Data loading method	For more details, see:
Custom	Creating custom connections
ELK	Creating ELK connections
Kafka	Creating Kafka connections

Event grouping and Probable cause

The three alert grouping algorithms and the Probable cause algorithm require event data. In the case of the Topological grouping algorithm and the Probable cause algorithm, topological data is also required. This topological data must be correlated with the event data.

Event data

In some cases, your event data is loaded into IBM Cloud Pak® for AIOps from a IBM® Netcool® Operations Insight®-based data source by using the following Netcool Operations Insight connector.

Data loading method	For more details, see:
Connecting an external ObjectServer	Creating IBM® Tivoli® Netcool®/OMNIbus connections

You can also load event data into the system by using other connection types. For more information, see Incoming connections.

Topological data

Load topology data into IBM Cloud Pak® for AIOps by configuring observers. For more information, see Defining observer jobs.

Preparing the AI environment

Prepare the AI environment by configuring performance and other setttings for the various AI algorithms.

For more information about preparing the AI environment, see the following topics:

- Adjusting model training resources
- Configuring alert expiry time
- Improving data streaming capacity

- About Elasticsearch indices and shards
- Managing embedded JSON
- Scaling log anomaly detection pods

How to modify training resources

You can modify training resources, such as to alter requirements.

Procedure

1. Find the custom resource of the algorithm for which you wish to alter the default resource requirements by running the following command:

   ```
   oc get algorithms
   ```

2. This should return something similar to the following:

   ```
   oc get algorithms
   ```

   Output:

   ```
   NAME                 AGE
   change-risk          46h
   log-anomaly          46h
   similar-incidents    46h
   ```

3. These CRs contain all the algorithm specifications. You can go ahead and edit the specific algorithm by running:

   ```
   oc edit algorithm <name>
   ```

4. Then you decode the *spec.manifestBase64* field. Once decoded you see a resources field defined as follows:

   ```
      resources:
         requests:
            cpu: 2
            memory: 4Gi
         limits:
            cpu: 2
            memory: 4Gi
   ```

   Note: You can edit requests and limits (For training algorithms, make sure that the **limits=requests**), re-encode to base64, and then replace the *manifestBase64* field in the CR with this newly encoded manifest.

5. Save the changes, and proceed with your training.

Configuring expiry time for log anomaly alerts

By default, alerts that are generated by a log anomaly expire in six hours (21,600 seconds). You can modify this value.

1. Access the IBM Cloud Pak Administration panel by following the instructions at UI console.
2. In the main navigation menu, click **Workloads** > **Deployments**.
3. Search by name, and type **log-anomaly** in the field. The **aimanager-aio-log-anomaly-detector** deployment is listed in the table.
4. Click **aimanager-aio-log-anomaly-detector** to visualize deployment details.
5. Click the **Environments** tab. A list of environment variables is displayed.
6. From the list, click **Add more** to add another environment variable.
7. Name the environment variable **EXPIRY_SECONDS**.
8. Set the value of the **EXPIRY_SECONDS** environment variable in seconds. For example, if you want to increase the expiry time to 24 hours, which is 86,400 seconds, set the value to `86400`.

9. Click **Save** to save the changes. This action triggers a restart of the **aimanager-aio-log-anomaly-detector** pod, which takes a few minutes.

After the pod restarts, the new expiry time applies to any alerts that are generated by a log anomaly.

Increasing data streaming capacity

If you have a production-sized deployment, you can increase data streaming throughput by increasing the number of Flink task managers.

About this task

You can increase the data throughput capacity for all of your incoming connections by increasing the **Base parallelism** number that is associated with that connection. The recommended value for source parallelism is equal to the number of days. For example, if you are pulling historical data for training for 7 days, make the source parallelism 7 or higher so that you can pull data for multiple days simultaneously.

Similarly, for base parallelism, use a higher value than 1 so that you can process data in parallel. In a small environment, the number of available Flink slots is 16. For a large environment, the maximum available number of slots is 32.

Table 1. Base parallelism for deployment sizes

Deployment size	Number of Flink task managers	Total number of base parallelisms available
Starter	2	16
Production	4	32

For more information about IBM Cloud Pak® for AIOps deployment size, see Hardware requirements.

When you define incoming log connections, increasing the base parallelism can improve the throughput or processing speed of an integration or both. If you define a live connection for anomaly detection, consider setting a base parallelism of 3 to 4 per 1,000 log messages per second. For example, if you are creating a connection to handle 8,000 log messages per second, a parallelism between 24 and 32 is recommended. Smaller base parallelisms reduce the potential need for you to scale your Flink task managers and increase your hardware resources.

Before you begin

- Determine whether your deployment is out of capacity. Run the following command from the namespace where IBM Cloud Pak for AIOps is deployed.

  ```
  oc exec -it cp4waiops-eventprocessor-eve-29ee-ep-jobmanager-0 -c jobmanager -- curl -k -u `oc get secret $(oc get secrets | grep cp4waiops-eventprocessor-eve-29ee-ep-admin-user | awk '!/-min/' | awk '{print $1;}') -o jsonpath="{.data.username}"|base64 -d`:`oc get secret $(oc get secrets | grep cp4waiops-eventprocessor-eve-29ee-ep-admin-user | awk '!/-min/' | awk '{print $1;}') -o jsonpath="{.data.password}"|base64 -d` https://localhost:8081/overview
  ```

  Sample output:

  ```
  {"taskmanagers":4,"slots-total":32,"slots-available":1,"jobs-running":2,"jobs-finished":0,"jobs-cancelled":1,"jobs-failed":0,"flink-version":"1.13.0","flink-commit":"f06faf1"}
  ```

- Check the logs for the task manager pods to see if an integration is working correctly and collecting live data.

  ```
  for pod in `oc get pods -n <namespace> | grep 'cp4waiops-eventprocessor-eve-29ee-ep-taskmanager' | awk '{ print $1 }' ` ; do oc logs $pod -n <namespace> | grep 'HttpSource'; done
  ```

  Where **<namespace>** is the namespace where IBM Cloud Pak for AIOps is deployed.

  Sample output:

```
2021-11-19 20:36:06,397 INFO  zeno.common.sources.HttpSource
[] - Results from extractor: 79932, total accumulated: 13769635, start time: 2021-11-
19T20:35:55.716Z, end time: 2021-11-19T20:36:06.396Z, elapse time in seconds: 10
2021-11-19 20:36:06,397 INFO  zeno.common.sources.HttpSource
[] - Querying window: 2021-11-19T20:35:52.740Z - 2021-11-19T20:36:02.740Z, window size:
10, time delay: 0s
2021-11-19 20:36:15,996 INFO  zeno.common.sources.HttpSource
[] - Results from extractor: 80247, total accumulated: 13849882, start time: 2021-11-
19T20:36:06.397Z, end time: 2021-11-19T20:36:15.996Z, elapse time in seconds: 9
2021-11-19 20:36:15,996 INFO  zeno.common.sources.HttpSource
[] - Querying window: 2021-11-19T20:36:02.741Z - 2021-11-19T20:36:12.741Z, window size:
10, time delay: 0s
2021-11-19 20:36:26,399 INFO  zeno.common.sources.HttpSource
[] - Results from extractor: 80280, total accumulated: 13930162, start time: 2021-11-
19T20:36:15.996Z, end time: 2021-11-19T20:36:26.399Z, elapse time in seconds: 10
2021-11-19 20:36:26,399 INFO  zeno.common.sources.HttpSource
[] - Querying window: 2021-11-19T20:36:12.742Z - 2021-11-19T20:36:22.742Z, window size:
10, time delay: 0s
```

In this example, the integration is behind in collecting live data. This example uses a querying window of **2021-11-19T20:36:12.742Z - 2021-11-19T20:36:22.742Z** at timestamp **2021-11-19 20:36:26,399**. The integration is approximately 4 seconds behind and is querying through **20:36:22** at a current time of **20:36:26** seconds. Increasing the base parallelism might improve results.

Procedure

Complete the following steps to increase the number of task managers on IBM Automation foundation Flink.

1. Access the Red Hat® OpenShift® command line and log in to your cluster.

   ```
   oc login -u kubeadmin -p <password> <URL for the Kubernetes/openshift API> --insecure-
   skip-tls-verify=true
   ```

2. Switch to the namespace where IBM Cloud Pak® for AIOps is installed.

   ```
   oc project <cp4waiops_namespace>
   ```

3. For Flink, run the following command before you increase the number of task managers:

   ```
   oc get pods | grep cp4waiops-eventprocessor-eve-29ee-ep-taskmanager
   ```

   The command output resembles the following example:

   ```
   cp4waiops-eventprocessor-eve-29ee-ep-taskmanager-0                1/1     Running     0
   10d
   cp4waiops-eventprocessor-eve-29ee-ep-taskmanager-1                1/1     Running     0
   10d
   ```

4. Update the number of task managers to the required value.

   By default, the value is **2** in a starter deployment, but you can increase the value to **3**. In a production deployment, the default value is **4**, but you can increase it to **5** if the node has sufficient resources.

   1. Run the following command to view and edit your subscription.

      ```
      oc edit subscription.operators.coreos.com ibm-aiops-orchestrator -n <namespace>
      ```

      Where **<namespace>** is the namespace where IBM Cloud Pak for AIOps is deployed.

   2. If there is not a **FLINK_TASK_MGR_REPLICAS** section, then add the following YAML, just beneath **spec**. Set the number of **FLINK_TASK_MGR_REPLICAS** to the required value, and save your changes.

      ```
      config:
        env:
        - name: FLINK_TASK_MGR_REPLICAS
          value: "3"
      ```

      Example:

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: ibm-aiops-orchestrator
  namespace: <namespace>
spec:
  config:
    env:
    - name: FLINK_TASK_MGR_REPLICAS
      value: "3"
  channel: v4.2
  installPlanApproval: Automatic
  name: ibm-aiops-orchestrator
  source: ibm-operator-catalog
  sourceNamespace: openshift-marketplace
```

Where **<namespace>** is the namespace where IBM Cloud Pak for AIOps is deployed.

5. For Flink, run the following command after you increase the number of replicas to **3**:

```
oc get pods | grep cp4waiops-eventprocessor-eve-29ee-ep-taskmanager
```

The command output resembles the following example:

```
cp4waiops-eventprocessor-eve-29ee-ep-taskmanager-0                1/1     Running     0
10d
cp4waiops-eventprocessor-eve-29ee-ep-taskmanager-1                1/1     Running     0
10d
cp4waiops-eventprocessor-eve-29ee-ep-taskmanager-2                1/1     Running     0
2m44s
```

About indices and shards

When setting up AI training, a critical factor is the number of shards available to support the creation of Elasticsearch indices. By default, a production (large) installation provides 1000 shards for use by the various AI training types and by the AI platform itself.

Each Elasticsearch index requires a certain number of shards. If the number of Elasticsearch indices used by the various AI training components requires more than the default of 1000 available shards, then some of the AI training might fail. A training failure returns a **Needs improvement** data qualtiy level, as described in Generating an AI model. In order to avoid training failure, you can do one or more of the following:

- Monitor the number of indices and shards currently in use and plan your AI training accordingly. For more information, see Determining your current shard consumption.
- Increase the number of shards available. Note that this will require provisioning of extra memory. For more information, see Increasing the number of Elasticsearch shards.

To avoid the possibility of running out of shards, a periodic cron job utility runs every hour to delete all the logtrain elastic indices that have an index creation date older than 14 days from the current timestamp.

Determining your current shard consumption

Monitor the number of indexes and shards currently in use so that you can plan your AI training accordingly.

About this task

When log data is collected for IBM Cloud Pak® for AIOps, the data is persisted within Elasticsearch for training purposes. While the collection of historical or live data for training is occurring or after the data collection completes, you can connect to Elasticsearch to check the progress and look at the indexes.

Connect to Elasticsearch

1. Open two console sessions on your computer.
2. Log in to your Red Hat® OpenShift® Container Platform cluster where IBM Cloud Pak for AIOps is installed.
3. In one console session, switch to the namespace where IBM Cloud Pak for AIOps is installed.
4. After the **oc login** command, run the following command to the cluster in one console. With this command, you port forward to the Elasticsearch service to access the service for checking the indexes.

   ```
   while true;
   do oc port-forward statefulset/$(oc get statefulset | grep iaf-system-elasticsearch-es-aiops | awk '{print $1}') 9200;
   done
   ```

5. In your second console session, run the following commands to obtain your username and password credentials for accessing Elasticsearch:

   ```
   export username=$(oc get secret $(oc get secrets | grep <instance>-elastic-secret | awk '!/-min/' | awk '{print $1;}') -o jsonpath="{.data.username}"|base64 -d)
   ```

   ```
   export password=$(oc get secret $(oc get secrets | grep <instance>-elastic-secret | awk '!/-min/' | awk '{print $1;}') -o jsonpath="{.data.password}"|base64 -d)
   ```

   Where **<instance>** is the name of your IBM Cloud Pak for AIOps instance, for example **ibm-cp-aiops**.

6. In the second console session, list the indexes that are stored in Elasticsearch. Look for index files that have a **1000-1000-<date>-logtrain** name format, such as **1000-1000-20210510-logtrain**. Use the following command to list the indexes in your cluster:

   ```
   curl -k -u $username:$password -XGET https://localhost:9200/_cat/indices | sort
   ```

   The following example illustrates a typical line of output:

   ```
   green open 1000-1000-20220302-logtrain        eEvaBufQ06yG074hmeHKG  8 1 244 6 72.7kb 72.7kb
   ```

   The following table lists the elements in this line of output:

Table. elements

Element	Description	Example
Index health	Health of the index.	green
Index Status	Status of the index.	open
Index name	Name of the index.	1000-1000-20220302-logtrain
UUID	Universal unique identifier for this index.	eEvaBufQ06yG074hmeHKG
Number of shards	Number of shards required by default for this index. This value varies based on the type of index. For example, an index that is associated with a day of ingested log data for natural language log anomaly training requires 8 shards for both a large and a small installation.	8
Number of replicas	Number of replicas required by default for this index. This value varies based on the type of index and the size of the installation. For example, an index that is associated with a day of ingested log data for natural language log anomaly training requires 1 replica for a large installation and 0 replicas for a small installation. A replica value of 1 replicates all of the shards for that index, meaning that a shard replica setting of 8/1 assigns 16 shards to the index.	1
docs.count	Detailed statistic associated with this index.	244
docs.deleted	Detailed statistic associated with this index.	6
Store size	Detailed statistic associated with this index.	72.7 kb
Primary store size	Detailed statistic associated with this index.	72.7 kb

7. Check the count of the records for an index by using the following command:

```
curl -k -u $username:$password -XGET https://localhost:9200/<indexname>/_count --insecure

{"count":318713,"_shards":{"total":1,"successful":1,"skipped":0,"failed":0}}
```

8. You can also look at sample data by using the following command:

```
curl -k -u $username:$password -XGET https://localhost:9200/<indexname>/_search --
insecure
```

Shard consumption in a large installation

Use the following table to help interpret the results returned. All values that are provided are for a production (large) installation.

Table. shard consumption

N.	AI component	Index name	Shard usage	Notes
	Natural language log anomaly detection (total)		Varies	
1	Each day of ingested log data	`1000-1000-`*date*`-logtrain`	16	The index for each extra day of data uses 16 shards.
2	Each model version total	`1000-1000-v_`x-contenttype	50	The index for each extra model version uses 50 shards.
3	Dedicated index for saving training data	`1000-1000-windowed_logs`	5	
4	Dedicated index for deploying models	`1000-1000-log_modles_latest`	5	
	Statistical baseline log anomaly detection (total)		20	
5	Statistical baseline log anomaly detection	`1000-1000-log_models_latest`	5	
6	Statistical baseline log anomaly detection	`1000-1000-reference_embeddin g`	5	
7	Statistical baseline log anomaly detection	`1000-1000-reference_oob`	5	
8	Statistical baseline log anomaly detection	`1000-1000-si_models_latest`	5	
9	**Change risk**	`1000-1000-cr_models_latest`		
10	**Similar tickets**	`1000-1000-si_models_latest`	10	
	Lifecycle services		Varies	
11	Each service	`aiops-searchservice-v10` *datetime*	5	The index for each extra lifecycle service uses 5 shards.
	AI platform		12	
12	AI platform	`postchecktrainingdetails0`	2	
13	AI platform	`prechecktrainingdetails0`	2	
14	AI platform	`trainingdefinition`	2	
15	AI platform	`trainingrun`	2	
16	AI platform	`trainingrunning`	2	
17	AI platform	`trainingstatus`	2	

Best practices

An analysis of the table in Shard consumption in a large installation shows that the most effective way to keep shard usage down is to follow these practices:

- Train natural language log anomaly detection on a smaller number of days. As you can see from the table row that is labeled 1, the index for each extra day of data uses 16 shards. The recommended number of days for training is 14.

  Note To avoid the possibility of running out of shards, a periodic cron job utility runs every hour to delete all the logtrain elastic indexes that have an index creation date older than 14 days from the current timestamp.

- Regularly delete new versions of natural language log anomaly detection models. As you can see from the table row that is labeled 2, the index for each extra model version uses 50 shards.

- Decrease the number of shards that are used by the index that is associated with each day of natural language log anomaly detection training. For more information, see Changing the number of shards per logtrain index.

  Note This can have an impact on Elasticsearch performance. In particular, writing to and reading from Elasticsearch can degrade. Note also that only training jobs created after you change this setting will assign the new shard value.

- When you define a log integration, such as for Elasticsearch, Logstash, and Kibana (ELK), Humio, or LogDNA, the Log Ops UI in IBM Cloud Pak for AIOps doesn't show the start and end date time zone when you use the **Historical data for initial AI training** data flow mode. You can't use the Log Ops UI to enter the start and end times. Instead, you can specify only the start and end dates. The start and end date time zone is Greenwich mean time (GMT).

Increasing the number of Elasticsearch shards

If you want to train your natural language log anomaly detection model on more than 14 days of log data, then you must increase the number of Elasticsearch shards.

About this task

Log anomaly training data is stored by timestamp. If data is missing, the number of days specified for historical training exceeds the processing capacity of the Elasticsearch nodes or pods.

Note: Implementing a longer data retention period requires extra resources. Additional CPU and memory is required for each additional Elasticsearch node or pod.

Procedure

If the log anomaly training for a data set requires more than 14 days, temporarily increase the number of shards by completing the following steps:

1. Run port-forwarding for Elasticsearch

   ```
   for i in {1..100}; do oc port-forward $(oc get po|grep iaf-system-elasticsearch-es-aiops-0|awk '{print $1;}'|head -n 1) 9203:9200; done
   ```

2. Get the Elasticsearch credentials:

   ```
   export username=$(oc get secret $(oc get secrets | grep "iaf-system-elasticsearch-es-default-user" | awk '{print $1;}') -o jsonpath="{.data.username}"|base64 -d) \
   export password=$(oc get secret $(oc get secrets | grep "iaf-system-elasticsearch-es-default-user" | awk '{print $1;}') -o jsonpath="{.data.password}"|base64 -d)
   ```

3. Set the maximum number of shards per node:

   ```
   curl -k -XPUT "https://$username:$password@localhost:9203/_cluster/settings" -H 'Content-Type: application/json' -d'
   {
       "transient": {
   ```

```
        "cluster.routing.allocation.total_shards_per_node": 2000
    }
}'
```

4. To support more shards, the memory must be provisioned accordingly. Otherwise, the shards won't be assigned to a worker node. For more information, see Size your shards in the Elasticsearch documentation.

   To increase the memory request and limit, and the Java virtual machine (JVM) size vertically, complete the following steps:

   - Find out the automationbase name:

     ```
     get automationbase
     NAME                    READY
     automationbase-sample   True
     ```

   - Run the following command with the name from the previous command:

     ```
     oc edit automationbase automationbase-sample
     ```

   - To increase the memory vertically, increase the `requests.memory`, `limits.memory`, and `ES_JAVA_OPTS` values. To increase the memory horizontally, increase the `replicas` value.

   Note: For a small installation, the High Availability (HA) for Elasticsearch is not enabled, so new pods run into the CrashloopBackOff. To enable HA, scale down the number of pods to zero by running the following command:

   ```
   ``` {: codeblock}
   oc scale statefulsets iaf-system-elasticsearch-es-aiops --replicas 0
   ```
   ```

   The existing pods are restarted and Elasticsearch will be in a good state.

Changing the number of shards per logtrain index

A logtrain index is created for each day of log data used by natural language log anomaly detection training. You can change the number of shards used by each logtrain index.

About this task

One way to keep overall shard usage down and thereby decrease the possibility of training failure is to decrease the number of shards used by each logtrain index.

Procedure

1. Access the OpenShift command line, and log in to your cluster.

   ```
   oc login -u kubeadmin -p <password> <URL for the Kubernetes/openshift API> --insecure-skip-tls-verify=true
   ```

2. Obtain the `eventprocessor` instance name under the eventprocessor project (namespace) of IBM Cloud Pak® for AIOps.

   ```
   oc get eventprocessor
   ```

3. Edit the yaml file for the `eventprocessor` instance by running the following command:

   ```
   oc edit eventprocessor cp4waiops-eventprocessor
   ```

4. Under the `spec.flink.env` setting, set the following environment variables to the desired settings:

   - `ES _NUM_SHARDS`: By default this is set to 8 shards.
   - `ES _NUM_REPLICAS`: By default this is set to 1 repllca for a large environment, and 0 replicas for a small environment.

```
spec:
 flink:
   authentication: {}
   env:
   - name: CARTRIDGE_NAME
     value: cp4waiops-cartridge
   - name: ES _NUM_SHARDS
     value: 8
   - name: ES _NUM_REPLICAS
     value: 1
```

5. Save the `yaml` file.

   Wait a few minutes to allow the relevant pods to restart.

Managing embedded JSON

Log lines containing embedded JSON might not be parsed correctly by the natual language log anomaly detection algorithm. If your log lines contain embedded JSON, then apply the methods described here to convert your log lines to unstructured data.

The following snippet shows an example of a log line containing embedded JSON.

```
021-09-23T21:43:40.067+0000  I NETWORK  [conn5083706]  received client metadata from
127.0.0.1:49736  conn5083706: { name: \"PyMongo\",  version: \"3.8.0\", type: \"Linux\",
architecture: \"x86_64\",  version: \"3.10.0-1160.25.1.el7.x86_64\", platform:
\"CPython3.7.3.final.0\" }
```

The braces (curly brackets) and the section between those braces is embedded JSON.

You can handle these log lines by applying one of the following options:

- **None**: The default option. The JSON is not processed or modified. This option results in the example JSON resembling the following code:

  ```
  021-09-23T21:43:40.067+0000  I NETWORK  [conn5083706]  received client metadata from
  127.0.0.1:49736  conn5083706:  { name: \"PyMongo\",  version: \"3.8.0\", type: \"Linux\",
  architecture: \"x86_64\",  version: \"3.10.0-1160.25.1.el7.x86_64\", platform:
  \"CPython3.7.3.final.0\" }
  ```

- **Flatten**: This option flattens the JSON object by removing the opening and closing braces. This option results in the example JSON resembling the following code:

  ```
  021-09-23T21:43:40.067+0000  I NETWORK  [conn5083706]  received client metadata from
  127.0.0.1:49736  conn5083706:  name: \"PyMongo\",  version: \"3.8.0\", type: \"Linux\",
  architecture: \"x86_64\",  version: \"3.10.0-1160.25.1.el7.x86_64\", platform:
  \"CPython3.7.3.final.0\"
  ```

- **Filter**: This option extracts the JSON object and replaces it with an empty string. This option results in the example JSON resembling the following code:

  ```
  021-09-23T21:43:40.067+0000  I NETWORK  [conn5083706]  received client metadata from
  127.0.0.1:49736  conn5083706:
  ```

  Note: Examine your data before applying this option, especially if your log lines are complete JSON objects.

Which option to select

Perform the following steps to get an idea of which option to select:

1. Retrieve a sample of your log data of at least 30,000 lines.
2. In your sample, determine the percentage of log lines that contain either JSON objects, or lines made up of both text and JSON objects.
3. Use the following table to help you select an option.

JSON percentage in data	Suggested option
Less than 30% of log lines contain JSON objects	**none**
More than 30% of log lines contain JSON objects	**filter**
More than 80% of log lines contain JSON objects	**none** or **flatten**

Limitations

Be aware of the following limitations:

1. Neither the **flatten** nor **filter** option checks the validity of the JSON object.
2. In some cases log data can include partial JSON. In this case, the **filter** options will extract the message preceding the first opening brace of the partial JSON object.

Selecting an option

To select a JSON processing option, see Adding a connection.

Scaling log anomaly detection pods

You can scale up the number of log anomaly detection pods for your instace.

Each **`log-anomaly-detection`** replica supports roughly 10 components. By default, a starter deployment has 2 **`log-anomaly-detection`** pods, while a production deployment has 4.

To scale the replicas to handle processing for additional components, modify the **`AIManagerMainProd`** custom resource.

Before you begin

Clusters can be for starter or production deployments, as described in Hardware requirements. A starter deployment cluster has limited resources, so you might want to check your cluster resources before you complete the following task to ensure that you are able to allocate extra resources.

Procedure

1. Log in to your cluster by running the **`oc login`** command.

   ```
   oc login -u kubeadmin -p <password>
   ```

   For more information, see Logging in to the OpenShift CLI.

2. Edit the IBM Cloud Pak for AIOps customer resource. The following command opens the file in the **`vi`** editor for editing.

   ```
   oc edit AIManagerMainProd aimanager
   ```

3. Search for the following configuration:

   ```
   spec:
     helmValues:
       global:
         logAnomaly:
           replicas: 2
   ```

4. Modify the replica count, such as to change the number of replicas to **`4`**.

   ```
   spec:
     helmValues:
       global:
   ```

```
      logAnomaly:
        replicas: 4
```

Note: Each instance of the `log-anomaly-detector` pod requires an additional 0.5 CPU cores and 0.5Gi of memory.

5. Save the file.
6. Restart the `ai-platform-api-server` pod to put the changes into effect. Restart the pod by deleting the associated pod. A new pod automatically starts up and uses the new parameters that are specified in the configmap.
    1. First, determine the name of the currently running pod.

        ```
        oc get po | grep ai-platform-api-server
        ```

        Your output can resemble the following sample output:

        ```
        aimanager-aio-ai-platform-api-server-7d75768cbc-zx5dh              1/1     Running
        0          6d1h
        ```

        Where,

        - The string `aimanager-aio-ai-platform-api-server-7d75768cbc-zx5dh` is the pod ID. The character string `7d75768cbc-zx5dh` uniquely identifies the pod. This string is the value for `<unique pod reference>` in the following steps.
        - The other information indicates that the pod is running and has been running for 6 days and 1 hour.
    2. Delete the pod by running the following command:

        ```
        oc delete po aimanager-aio-ai-platform-api-server-7d75768cbc-zx5dh
        ```

    3. Wait a short time, then run the following command to check that the new pod started up, and to show the new pod ID.

        ```
        oc get po | grep ai-platform-api-server
        ```

Configuring log formats for WebSphere log anomaly detection

To access log messages for WebSphere log anomaly detection, configure the domains with the correct type of log formats.

- Configure IBM WebSphere Application Server to use High Performance Extensible Logging (HPEL).
    - HPEL provides a mechanism for storing and accessing log, trace, `System.err`, and `System.out` information that is generated by the server or generated by applications that use `java.util.logging`.
    - To render data stored by HPEL in JSON format, use the LogViewer command.
- Configure IBM WebSphere Liberty servers to output JSON to the `messages.log` file.
- Only log messages from IBM WebSphere Liberty server code have IBM message IDs, so IBM Cloud Pak for AIOps can use only those messages. IBM Cloud Pak for AIOps can't run log anomaly detection on logs from applications that run on IBM WebSphere Application Server.
    - Thus, the upstream log data source for IBM Cloud Pak for AIOps is either HPEL for IBM WebSphere Application Server or a `messages.log` file for an IBM WebSphere Liberty server.

About AI algorithms

AI algorithms analyze different kinds of data and produce AI models, which automate research, reducing the need for manual work in solving problems. These unsupervised algorithms enable the use of unlabeled data for training AI models without any human intervention. IBM Cloud Pak® for AIOps training generates models for a number of different AI algorithms. You can train the models for each of these AI algorithms independently of each other.

You use the AI Model Management to set up training for various AI algorithms.

The AI Model Management has these tabs: **Training** and **Data assets**.

1. The **Training** tab allows you to view two types of algorithms in tile format:

    - **Trainable AI algorithms** These algorithms must be trained on certain data types before they can create deployable models. They include change risk, log anomaly detection-natural language, metric anomaly detection, similar tickets, and temporal grouping. These algorithms are listed as tiles. You can set up the training for each algorithm separately.

    - **Pre-trained AI algorithms** These algorithms don't require models to derive insights. They are enabled by default. These algorithms include probable cause, scope-based grouping, topological grouping, and log anomaly detection-statistical baseline. Again each of these algorithms is represented by a tile.

2. The **Data assets** tab shows the data connections that are feeding your AI training. It lists them in a table and gives details about each connection, such as Connection name, Data flow status (Running or Finished), Connection type, Data type, and Data collected (Live or Historical).

Note: It is worth noting that naming the training is an optional step in setting up training for trainable AI algorithms.

For more information about AI algorithms, see the following topics:

- About change risk
- About event grouping
- About log anomaly detection algorithms
- About metric anomaly detection
- About probable cause
- About similar tickets

About change risk

This AI algorithm provides an assessment of the risk of implementing a proposed change; for example, a code change or a new software version.

Description

Hundreds of changes can affect an application during its lifecycle. *Change risk* is an unsupervised learning algorithm that takes historical data about all of those changes and helps you determine how likely it is that a specific change would cause a problem, based on how successfully that similar change was deployed in the past.

Using the assessment score provided by this AI model, you can determine how safe it is to proceed with a change.

To train this algorithm, you must complete some setup tasks, as described in Setting up training for change risk.

When a *Change risk* algorithm is successfully trained and the resulting AI model deploys, if someone opens a change request ticket, a *Change risk* assessment appears in the proactive ChatOps channel (for more information, see Change risk in ChatOps). When the assessment is deemed to be high risk, then an alert is generated and is displayed in the Alert Viewer.

Note: Some training features are specific to the change risk algorithm. For more information, see Managing algorithms.

Prerequisites

To train a change risk algorithm, you need change request and incident ticket data from ServiceNow. Starting with IBM Cloud Pak® for AIOps version 4.1.0, similar incidents, such as from a ServiceNow integration, are called similar tickets.

Before you set up training for this algorithm, create a functioning data connection to ServiceNow. For more information, see Creating a ServiceNow connection.

Note: If you enable collection of historical data when you connect to ServiceNow, then data for the selected timeframe is available to train on.

To provide the best data for training, you need a mix of successful and failed change tickets. Around 10,000 change tickets, with a minimum 2% failure rate, is considered ideal. However, models can also be created with much less data.

Language support

For information about supported languages for this algorithm, see Language support.

About event grouping

Three AI algorithms are provided that group events together and present the groups within a single incident.

Description

The three algorithms are the following:

- Temporal grouping
- Topological grouping
- Scope-based Grouping

Temporal grouping

This AI algorithm is an unsupervised learning algorithm that groups alerts that are discovered to co-occur over time. When a problem arises, there are typically multiple parts of a system or environment that are impacted. When alerts in different areas co-occur, it makes sense to look at them together and treat them as one problem to try to determine what might have happened.

Grouping co-occurring alerts together reduces the number of tickets and incidents opened and the number of people looking at the same problem, thereby significantly reducing noise in your monitoring systems. It helps you to understand the context of an issue so you can prioritize, triage, and resolve it more quickly.

To train this algorithm, complete some initial setup tasks. This AI algorithm also needs at least three months of historical event data to learn alert patterns and discover groups. For more information, see Setting up training for temporal grouping.

This algorithm is enabled so when related alerts are grouped based on when they occur, Site reliability engineers (SREs) and other users responsible for application and service availability will be able to view the details in the event Viewer, as described in About alerts.

Topological grouping

This AI algorithm is an unsupervised learning algorithm that groups alerts based on the resource groups in which those alerts occur. For example if you have a resource group made up of all the resources within a given Kubernetes namespace, then any alerts on pods, microservices, or other resources in that namespace will be grouped together in a single topological group.

Topological grouping helps you understand when alerts are connected based on their topology, providing valuable context for why related alerts might occur together.

This algorithm is enabled so when related alerts are grouped based on their topology, Site reliability engineers (SREs) and other users responsible for application and service availability will be able to view the details in the Alert Viewer, as described in About alerts.

Note:

1. Topology resources in the same group do not need to be connected for alert grouping to happen.
2. Alerts that target topology resources in the same application (service) are also grouped.
3. Alerts must occur within a 15 minute time window to get correlated.

Topological grouping requires that there be a topological group creator.

Scope-based grouping

This AI algorithm is an unsupervised learning algorithm that automatically groups alerts relating to an incident if they have the same defined scope and occur during the same period of time. A scope can be used to identify where events originate based on a common attribute, for example, the location of a server room.

By understanding when alerts are related based on both time and location, you can more quickly diagnose incidents.

This algorithm is enabled so when related alerts are grouped based on their scope, Site reliability engineers (SREs) and other users responsible for application and service availability will be able to view the details in the Alert Viewer, as described in About alerts.

Prerequisites

The three event grouping algorithms require event data. In the case of the Topological grouping algorithm, topological data is also required. This topological data must be correlated with the event data.

For correlations, they must exist within the same topological group or template.

Event data

In most cases your event data will be loaded into IBM Cloud Pak® for AIOps from a IBM® Netcool® Operations Insight®-based data source, using one of the methods listed in the following table.

Data loading method	For more details, see:
Connecting on-premises probes	Connecting on-premises probes to the in-cluster ObjectServer
Connecting an external ObjectServer	Connecting an external ObjectServer into the system using an XML Gateway

You can also load event data into the system using the generic custom and Kafka methods.

- For more information on the custom connection, see Creating custom connections.
- For more information on the Kafka connection, see Creating Kafka connections.

Topological data

Load topology data into IBM Cloud Pak® for AIOps by configuring observers. For more information, see Defining observer jobs.

Language support

For information about supported languages for this algorithm, see Language support.

About log anomaly detection algorithms

IBM Cloud Pak® for AIOps processes incoming logs as part of the log anomaly detection process.

There are two log anomaly detection AI algorithms, each of which can run independently of the other. If both algorithms are enabled, then any log anomalies discovered by both will be reconciled, so that only one alert is generated. In this case, the severity of the combined alert will be equal to the highest severity of the two alerts.

Site reliability engineers (SREs) and other users responsible for application and service availability are able to display log anomalies as alerts within the context of an incident, as described in Viewing relevant events. They will also be able to view these log anomaly alerts in the Alert Viewer, as described in About alerts.

Prerequsites

Log data

Logs are sent from an application environment to a logging system, such as LogDNA or Splunk, through data shippers, such as Logstash. Each logging system has a mapping definition in Cloud Pak for AIOps.

The Log anomaly detection algorithms require log data from one or more of your log systems. Before attempting to set up training for these algorithms, ensure that you have created a functioning data connection to at least one of the log systems listed in the following table.

Table. Log system details

Log system	For more details, see:
Falcon LogScale	Creating Falcon LogScale connections
LogDNA	Creating LogDNA connections
Splunk	Creating Splunk connections

You can also load log data into the system using one of these generic data loading methods.

Table. Data loading method details

Data loading method	For more details, see:
Custom	Creating custom connections
ELK	Creating ELK connections
Kafka	Creating Kafka connections

Performance tuning

Before training can occur, logs are pulled from a logging system, such as LogDNA or Splunk, and are stored in Elasticsearch.

Before attempting to set up training for these algorithms, review the performance tuning tasks at the following location: Setting up training for log anomaly detection.

Language support

For information about supported languages for this algorithm, see Language support.

Algorithms

The two algorithms are as follows:

- Natural language
- Statistical baseline

About log anomaly detection - natural language

Natural language log anomaly detection is an unsupervised learning algorithm that uses Natural Language Processing (NLP) on a subset of your log data to discover abnormal behavior.

Log anomaly detection takes large amounts of log data and trains on it to learn what is normal behavior for a particular component. This model goes beyond just looking at error states or frequency of metadata around log messages. Instead, it determines when something becomes an anomaly compared to what patterns it typically exhibits.

To train this AI algorithm, you must complete some initial setup tasks, as described in Setting up training for log anomaly detection - natural language

For more information about natural language log anomaly detection, see the following topics:

- Workflow
- Examples

Workflow of natural language log anomaly detection

Find out how natural language log anomaly detection processes your log data to generate log alerts.

The workflow of natural language log anomaly detection is made up of two parts:

- Model generation
- Application of the model to the live log data stream

Model generation

The following figure shows a simplified workflow illustrating how the natural language log anomaly detection algorithm generates an AI model from log data. This part of the workflow is known as **training**.

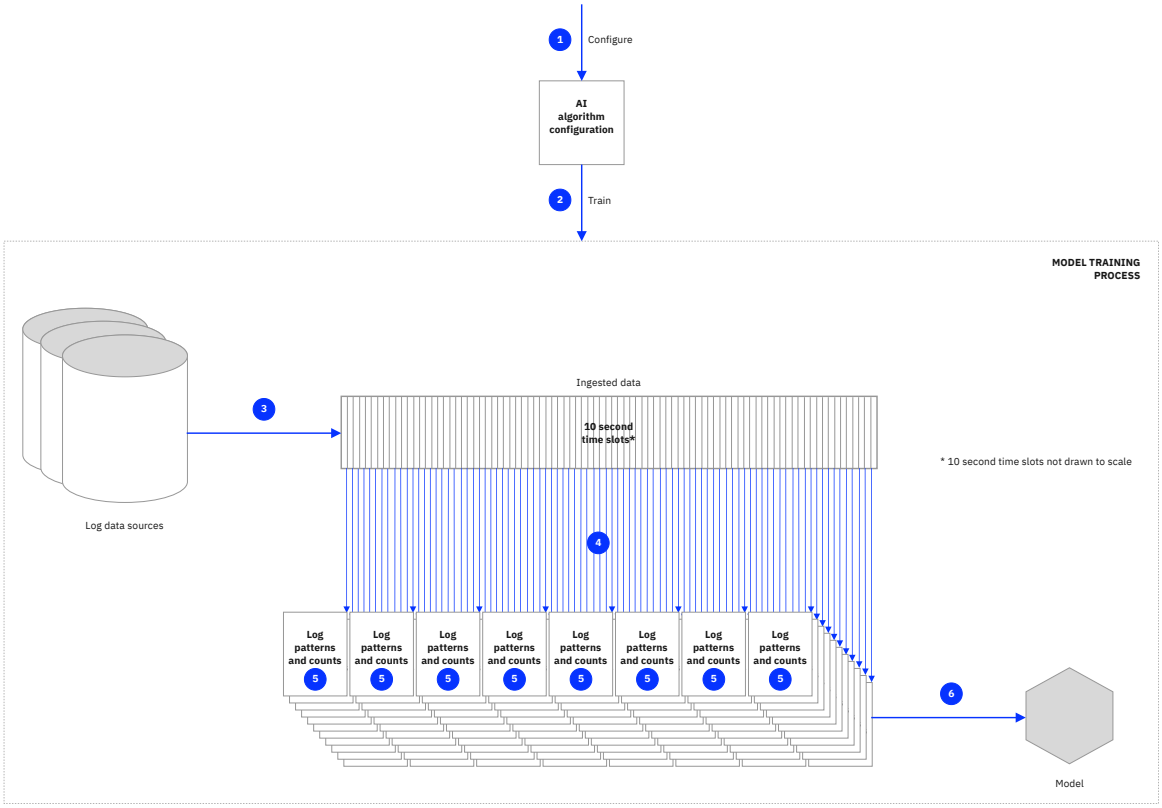


Figure. Workflow

1. **Algorithm is configured and trained**

   The algorithm configuration includes specification of log data sources, the period of data to train on, whether to run training on a schedule or on demand, and whether to deploy the training automatically or manually. Training is started and the model training process begins, as described in steps 2 to 6.

2. **Training jobs are identified**

   Assume that you have a microservice that is made up of multiple components such as orders, catalog, front-end, and more, each of these components with its own set of logs. As it is likely that each of the components has different logging behavior, the log anomaly detection algorithm creates a separate model for each component. Each of these models is trained by a separate training job.

   The logging data for the different components is differentiated by using the `instance-id` parameter that is specified during log connection mapping. For an example of log connection mapping for a LogDNA log system, see Specifying field mapping.

   The following steps 3 to 6 explain the process for training a unique model for each component. The steps are repeated for each component identified.

   Note: For more information about limits and throughput rates, see Supported resource number and throughput rates.

3. **Data is imported**

   Based on the specification of log data sources in the algorithm configuration, data is imported from log data sources for a given component.

4. **Data is divided into 10-second slots**

   The imported data is divided into 10-second time slots. For example, if two weeks' worth of data was impoirted, then that data is divided into x time slots based on the following calculation.

   2 weeks = 14 days = 336 hours = 20,160 minutes = 120,960 time slots

5. **Log patterns are identified and counted**

   Within each time slot, the system identifies and counts log patterns. Log patterns are identified in the following way:

   1. Each log message is separated into variant and invariant parts. For example, consider the following log messages:

```
2021-03-12T03:05:31.608355778+00:00 stdout F
{
  "name":"@instana collector",
  "__in":5,
  "hostname":"cart-846b9595c9qcqfk",
  "pid":35261,
  "module": "announceCycle/agentHostLookup",
  "level":2,
  "msg":"Agent cannot be contacted through system ABC nor through default gateway
123. Scheduling reattempt 0
}

2021-03-12T03:05:33.608355778 +00:00 stdout F
{
  "name":"@instana collector",
   "__in":5,
   "hostname":"cart-846b9595c9qcqfk",
   "pid":89561,
   "module": "announceCycle/agentHostLookup",
   "level":2,
   "msg":"Agent cannot be contacted through system DEF nor through default gateway
456. Scheduling reattempt 1
 }

2021-03-12T03:05:35.321654987+00:00 stdout F
{
 "name":"@instana collector",
 "__in":5,
 "hostname":"cart-846b9595c9qcqfk",
 "pid":75421,
 "module": "announceCycle/agentHostLookup",
 "level":3,
 "msg":"Agent cannot be contacted through system GHK nor through default gateway
789. Scheduling reattempt 2
}
```

      These log messages can be expressed by using the following log pattern, where the variant parts of the log messages are expressed by using wildcards, such as **<*>** and **<NUM>**.

```
<*> stdout F
{
  "name":"@instana collector",
  "__in":<NUM>,
  "hostname":"cart-846b9595c9qcqfk",
  "pid":<NUM>,
  "module": "announceCycle/agentHostLookup",
  "level":<NUM>,
  "msg":"Agent cannot be contacted through <*> nor through default gateway <*>.
Scheduling reattempt <NUM>
}
```

   2. Log patterns are counted in each 10-second slot and means and standard deviations are stored for each log pattern.

6. **Model is created**

   On completion of training, a model is created that includes log pattern statistics.

   Note: Each time that the algorithm is run, steps 2 to 5 are repeated from scratch and a new version of the model is generated.

Application of the model to the live log data stream

The following figure shows a simplified workflow illustrating how the natural language log anomaly detection algorithm applies the AI model to the live log data stream. This part of the workflow is known as **inference**, and it involves applying the findings from the model.

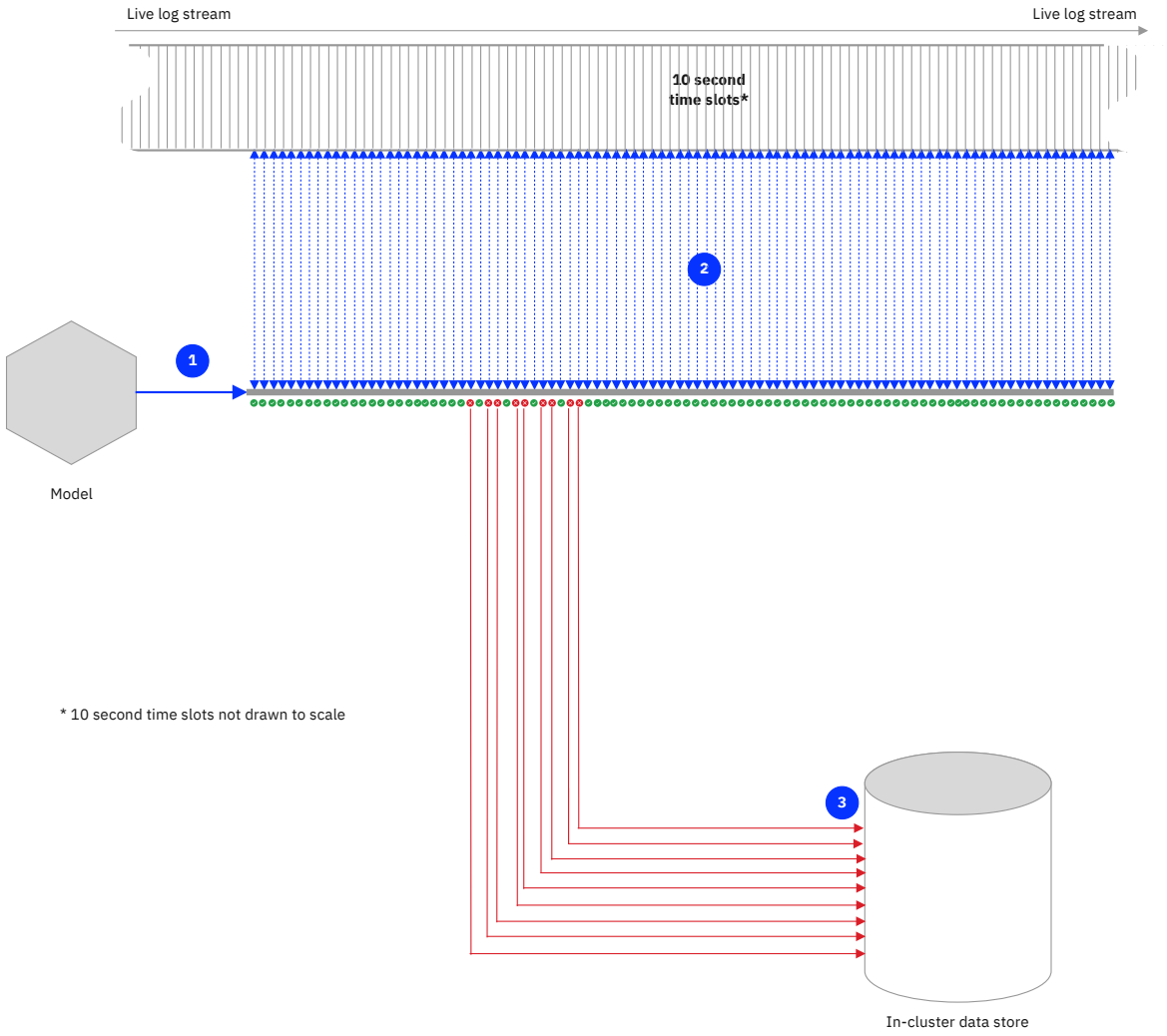


Figure. Workflow model generation

1. **Model is deployed**

   Depending on how the algorithm was configured a new version of the model is deployed automatically on completion of training, or manually.

2. **Review live stream for anomalies**

   Now that you have a deployed model, the system starts monitoring the live log stream for anomalies, which are statistically significant deviations from the baseline. It evaluates each component's logs every 10 seconds for anomalies.

3. **Log alerts are generated**

Log anomalies are generated at the end of each 10-second interval for each log line in which there is a statistical difference between the log pattern counts and the reference values and log patterns – or both – are identified in the live log stream that does not exist in the model.

Based on this analysis, the log anomaly detection algorithms identify one or more of the following log anomaly types:

1. Pattern is expected but is not detected
2. Pattern is detected but is not expected
3. Pattern is detected but is not expected, and includes an error
4. Pattern is detected with lower frequency than expected
5. Pattern is detected with higher frequency than expected

The algorithm creates a log alert for each log anomaly. The severity of the log alert is calculated based on the following considerations:

- All alerts that are generated by this algorithm are given an intial severity of 4 (Minor)
- If the pattern is detected but not expected (log anomaly type 2) or it is an error pattern, then the severity is increased to 5 (Major).
- If the pattern is detected with a lower frequency than expected (log anomaly type 4), then the severity is increased to 6 (Critical).

Example of natural language log anomaly detection

This topic provides an examples of log anomalies generated by the natural language log anomaly detection algorithm and when they are likely to be found.

Example: unexpected log pattern

You recently installed an application, which is made up of multiple microservices. Most of the microservices that make up this application generate their own log files.

Note For the natural language log anomaly detection algorithm to work, the following minimum data requirements must be met. At least 75% of the component microservices must have at least 2000 data points, where a data point is usually a log line.

The example proceeds as follows:

1. Data from the microservices is trained, as described in Workflow of natural language log anomaly detection.

2. During training, log patterns statistics are gathered and stored in the initial version of the model, as described in Workflow of natural language log anomaly detection. One of the patterns identified in the model is the following pattern within the **message** keyword.

   ```
   incident euid=<NUM> epid=<NUM> dsteuid=<NUM> dstepid=<NUM> <*>type=undefined
   subtype=undefined level=notice
   ```

   This pattern occurs on average roughly 2.5 times in every 10 second time slot. See Clean log file lines for an example of what a log like this might look like.

3. When the model is applied to the live data stream as described in Workflow of natural language log anomaly detection, at a certain point the log pattern described in the previous step starts occurring in a number of the 10 second time slots. See Error log file lines for an example of what a log like this might look like.

4. Log anomaly alerts are generated for each log anomaly and sent to the in-cluster ObjectServer. These alerts report that a pattern has been detected with higher frequency than expected. See Log anomaly alert format for an example of what a log anomaly alert might look like.

Reference information

The following information is referenced in this task.

- Clean log file lines

- Error log file lines
- Log anomaly alert example

Clean log file lines

These example JSON log snippets display log lines for one of the microservices within the application.

Clean log file lines

The pattern of interest is:

```
incident euid=<NUM> epid=<NUM> dsteuid=<NUM> dstepid=<NUM> <*>type=undefined subtype=undefined
level=notice
```

You can find this pattern in the **message** parameter within the log lines below by searching for the invariant code **type=undefined subtype=undefined level=notice**. This will show you that the pattern of interest only occurs once, in the first line.

In a real life example, the number of lines and the number of occurrences of the pattern in a 10 second time slot would be consistent with the two or three times that is expected based on the values in the model.


```
[
    {"_index":"filebeat-7.12.1-2021.08.20-
000001","_type":"_doc","_id":"zpprensBxt31inA6cZbF","_score":null,"_source":
{"@timestamp":"2021-08-24T23:08:00.283Z","stream":"stdout","input":
{"type":"container"},"container":
{"runtime":"docker","id":"bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53","i
mage":{"name":"registry.gitlab.com/quote-of-the-day/qotd-ratings-service:build-
354180024"}},"ecs":{"version":"1.8.0"},"log":{"offset":640306,"file":
{"path":"/var/log/containers/qotd-ratings-5444ddd79c-gp7kb_qotd_qotd-rating-
bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53.log"}},"message":"incident
euid=2570 epid=7324 dsteuid=2411 dstepid=7148 vd=Jarret10 type=undefined subtype=undefined
level=notice","kubernetes":{"container":{"name":"qotd-
rating","image":"registry.gitlab.com/quote-of-the-day/qotd-ratings-service:build-
354180024"},"deployment":{"name":"qotd-ratings"},"node":{"uid":"193183f9-01bd-11ec-926b-
0617a6095287","labels":{"node-
role_kubernetes_io/master":"true","beta_kubernetes_io/arch":"amd64","beta_kubernetes_io/os":"l
inux","kubernetes_io/hostname":"padme-okd.coc-ibm.com","node-
role_kubernetes_io/compute":"true","node-role_kubernetes_io/infra":"true"},"hostname":"padme-
okd.coc-ibm.com","name":"padme-okd.coc-ibm.com"},"namespace_uid":"dcc7d8e8-01bf-11ec-9802-
0617a6095287","pod":{"name":"qotd-ratings-5444ddd79c-gp7kb","uid":"d9b9549f-0529-11ec-9802-
0617a6095287"},"namespace":"qotd","replicaset":{"name":"qotd-ratings-5444ddd79c"},"labels":
{"pod-template-hash":"1000888357","app":"qotd-rating"}},"agent":{"hostname":"padme-okd.coc-
ibm.com","ephemeral_id":"39a50623-e967-451f-b30d-f06fd421d0cf","id":"456adf88-c87d-49d2-8aa2-
9b93d39ba057","name":"padme-okd.coc-ibm.com","type":"filebeat","version":"7.12.1"},"host":
{"name":"padme-okd.coc-ibm.com"}},"sort":[1629846480283,640306]}

    {"_index":"filebeat-7.12.1-2021.08.20-
000001","_type":"_doc","_id":"z5prensBxt31inA6d5Z9","_score":null,"_source":
{"@timestamp":"2021-08-24T23:08:02.065Z","agent":{"version":"7.12.1","hostname":"padme-
okd.coc-ibm.com","ephemeral_id":"39a50623-e967-451f-b30d-f06fd421d0cf","id":"456adf88-c87d-
49d2-8aa2-9b93d39ba057","name":"padme-okd.coc-ibm.com","type":"filebeat"},"log":
{"offset":11804878,"file":{"path":"/var/log/containers/qotd-web-5cbb5d9f64-srmc4_qotd_qotd-
web-
f6c3fb360307ff0759831d56124674a15af7d14e135a19928f4b3641aeee605c.log"}},"stream":"stdout","mes
sage":"Starting new request token: 783499 for IP: 10.128.0.41","input":
{"type":"container"},"kubernetes":{"container":{"name":"qotd-
web","image":"registry.gitlab.com/quote-of-the-day/qotd-web:latest"},"deployment":
{"name":"qotd-web"},"node":{"labels":{"kubernetes_io/hostname":"padme-okd.coc-ibm.com","node-
role_kubernetes_io/compute":"true","node-role_kubernetes_io/infra":"true","node-
role_kubernetes_io/master":"true","beta_kubernetes_io/arch":"amd64","beta_kubernetes_io/os":"l
inux"},"hostname":"padme-okd.coc-ibm.com","name":"padme-okd.coc-ibm.com","uid":"193183f9-01bd-
11ec-926b-0617a6095287"},"namespace_uid":"dcc7d8e8-01bf-11ec-9802-0617a6095287","pod":
{"uid":"efbf8ddd-04fc-11ec-9802-0617a6095287","name":"qotd-web-5cbb5d9f64-
srmc4"},"namespace":"qotd","replicaset":{"name":"qotd-web-5cbb5d9f64"},"labels":{"app":"qotd-
web","pod-template-hash":"1766185920"}},"container":
{"runtime":"docker","id":"f6c3fb360307ff0759831d56124674a15af7d14e135a19928f4b3641aeee605c","i
mage":{"name":"registry.gitlab.com/quote-of-the-day/qotd-web:latest"}},"ecs":
{"version":"1.8.0"},"host":{"name":"padme-okd.coc-ibm.com"}},"sort":[1629846482065,11804878]}
```

```
    {"_index":"filebeat-7.12.1-2021.08.20-
000001","_type":"_doc","_id":"0JprensBxt31inA6d5Z9","_score":null,"_source":
{"@timestamp":"2021-08-24T23:08:02.066Z","host":{"name":"padme-okd.coc-ibm.com"},"ecs":
{"version":"1.8.0"},"stream":"stdout","message":"[783499] Web request: /.","input":
{"type":"container"},"kubernetes":{"pod":{"uid":"efbf8ddd-04fc-11ec-9802-
0617a6095287","name":"qotd-web-5cbb5d9f64-srmc4"},"namespace":"qotd","replicaset":
{"name":"qotd-web-5cbb5d9f64"},"labels":{"pod-template-hash":"1766185920","app":"qotd-
web"},"container":{"name":"qotd-web","image":"registry.gitlab.com/quote-of-the-day/qotd-
web:latest"},"deployment":{"name":"qotd-web"},"node":{"hostname":"padme-okd.coc-
ibm.com","name":"padme-okd.coc-ibm.com","uid":"193183f9-01bd-11ec-926b-0617a6095287","labels":
{"kubernetes_io/hostname":"padme-okd.coc-ibm.com","node-
role_kubernetes_io/compute":"true","node-role_kubernetes_io/infra":"true","node-
role_kubernetes_io/master":"true","beta_kubernetes_io/arch":"amd64","beta_kubernetes_io/os":"l
inux"}},"namespace_uid":"dcc7d8e8-01bf-11ec-9802-0617a6095287"},"container":
{"runtime":"docker","id":"f6c3fb360307ff0759831d56124674a15af7d14e135a19928f4b3641aeee605c","i
mage":{"name":"registry.gitlab.com/quote-of-the-day/qotd-web:latest"}},"log":
{"offset":11805003,"file":{"path":"/var/log/containers/qotd-web-5cbb5d9f64-srmc4_qotd_qotd-
web-f6c3fb360307ff0759831d56124674a15af7d14e135a19928f4b3641aeee605c.log"}},"agent":
{"name":"padme-okd.coc-ibm.com","type":"filebeat","version":"7.12.1","hostname":"padme-
okd.coc-ibm.com","ephemeral_id":"39a50623-e967-451f-b30d-f06fd421d0cf","id":"456adf88-c87d-
49d2-8aa2-9b93d39ba057"}},"sort":[1629846482066,11805003]}

    {"_index":"filebeat-7.12.1-2021.08.20-
000001","_type":"_doc","_id":"0ZprensBxt31inA6d5Z9","_score":null,"_source":
{"@timestamp":"2021-08-24T23:08:02.182Z","input":{"type":"container"},"kubernetes":
{"deployment":{"name":"qotd-web"},"node":{"name":"padme-okd.coc-ibm.com","uid":"193183f9-01bd-
11ec-926b-0617a6095287","labels":{"node-role_kubernetes_io/compute":"true","node-
role_kubernetes_io/infra":"true","node-
role_kubernetes_io/master":"true","beta_kubernetes_io/arch":"amd64","beta_kubernetes_io/os":"l
inux","kubernetes_io/hostname":"padme-okd.coc-ibm.com"},"hostname":"padme-okd.coc-
ibm.com"},"namespace_uid":"dcc7d8e8-01bf-11ec-9802-0617a6095287","pod":{"uid":"efbf8ddd-04fc-
11ec-9802-0617a6095287","name":"qotd-web-5cbb5d9f64-srmc4"},"namespace":"qotd","replicaset":
{"name":"qotd-web-5cbb5d9f64"},"labels":{"app":"qotd-web","pod-template-
hash":"1766185920"},"container":{"name":"qotd-web","image":"registry.gitlab.com/quote-of-the-
day/qotd-web:latest"}},"host":{"name":"padme-okd.coc-ibm.com"},"agent":{"id":"456adf88-c87d-
49d2-8aa2-9b93d39ba057","name":"padme-okd.coc-
ibm.com","type":"filebeat","version":"7.12.1","hostname":"padme-okd.coc-
ibm.com","ephemeral_id":"39a50623-e967-451f-b30d-f06fd421d0cf"},"stream":"stdout","message":"
[783499] Getting daily quote.","log":{"file":{"path":"/var/log/containers/qotd-web-5cbb5d9f64-
srmc4_qotd_qotd-web-
f6c3fb360307ff0759831d56124674a15af7d14e135a19928f4b3641aeee605c.log"},"offset":11805098},"con
tainer":{"image":{"name":"registry.gitlab.com/quote-of-the-day/qotd-
web:latest"},"runtime":"docker","id":"f6c3fb360307ff0759831d56124674a15af7d14e135a19928f4b3641
aeee605c"},"ecs":{"version":"1.8.0"}},"sort":[1629846482182,11805098]}

    {"_index":"filebeat-7.12.1-2021.08.20-
000001","_type":"_doc","_id":"3pprensBxt31inA6h5Yj","_score":null,"_source":
{"@timestamp":"2021-08-24T23:08:02.195Z","kubernetes":{"namespace":"qotd","replicaset":
{"name":"qotd-quote-585979769"},"labels":{"app":"qotd-quote","pod-template-
hash":"141535325"},"container":{"name":"qotd-quote","image":"registry.gitlab.com/quote-of-the-
day/quote-service:latest"},"deployment":{"name":"qotd-quote"},"node":{"uid":"193183f9-01bd-
11ec-926b-0617a6095287","labels":{"node-role_kubernetes_io/compute":"true","node-
role_kubernetes_io/infra":"true","node-
role_kubernetes_io/master":"true","beta_kubernetes_io/arch":"amd64","beta_kubernetes_io/os":"l
inux","kubernetes_io/hostname":"padme-okd.coc-ibm.com"},"hostname":"padme-okd.coc-
ibm.com","name":"padme-okd.coc-ibm.com"},"namespace_uid":"dcc7d8e8-01bf-11ec-9802-
0617a6095287","pod":{"name":"qotd-quote-585979769-wpg25","uid":"e810e7e8-04fc-11ec-9802-
0617a6095287"}},"container":{"image":{"name":"registry.gitlab.com/quote-of-the-day/quote-
service:latest"},"runtime":"docker","id":"ef6bf31678a9d80b5ac95a27dbb72f68a2f855aa09874e6eda22
2e4ff1c3184e"},"log":{"file":{"path":"/var/log/containers/qotd-quote-585979769-
wpg25_qotd_qotd-quote-
ef6bf31678a9d80b5ac95a27dbb72f68a2f855aa09874e6eda222e4ff1c3184e.log"},"offset":6052045},"inpu
t":{"type":"container"},"agent":{"id":"456adf88-c87d-49d2-8aa2-9b93d39ba057","name":"padme-
okd.coc-ibm.com","type":"filebeat","version":"7.12.1","hostname":"padme-okd.coc-
ibm.com","ephemeral_id":"39a50623-e967-451f-b30d-f06fd421d0cf"},"ecs":
{"version":"1.8.0"},"host":{"name":"padme-okd.coc-ibm.com"},"stream":"stdout","message":"
[783499] Quote request: /daily."},"sort":[1629846482195,6052045]}
]
```


Error log file lines

These example JSON log snippets display log lines for one of the microservices within the application. You can see that the pattern of interest occurs in every line in this log snippet, in all five times.

The pattern of interest is:

incident euid=<NUM> epid=<NUM> dsteuid=<NUM> dstepid=<NUM> <*>type=undefined subtype=undefined level=notice

You can find this pattern in the **message** parameter within the log lines below by searching for the invariant code **type=undefined subtype=undefined level=notice**. This will show you which lines of code instantiate this pattern.

In a real life example, the number of lines and the number of occurrences of the pattern in a 10 second time slot would be significantly larger. You might have 100 or so log lines in 10 seconds and the pattern might occur on the order of 20 times, which is a significant deviation from the two or three times that is expected based on the values in the model. Based on this, log anomaly alerts would be generated.

Error log file lines


```
[
    {"_index":"filebeat-7.12.1-2021.08.20-
000001","_type":"_doc","_id":"pppqensBxt31inA6h5Qa","_score":null,"_source":
{"@timestamp":"2021-08-24T23:07:00.053Z","input":      {"type":"container"},"container":
{"id":"bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53","image":
{"name":"registry.gitlab.com/quote-of-the-day/qotd-ratings-service:build-
354180024"},"runtime":"docker"},"host":{"name":"padme-okd.coc-ibm.com"},"log":{"file":
{"path":"/var/log/containers/qotd-ratings-5444ddd79c-gp7kb_qotd_qotd-rating-
bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53.log"},"offset":610747},"strea
m":"stdout","message":"incident euid=5840 epid=9800 dsteuid=3612 dstepid=7788
vd=Corbin_Gerlach59 type=undefined subtype=undefined level=notice","kubernetes":
{"namespace_uid":"dcc7d8e8-01bf-11ec-9802-0617a6095287","pod":{"name":"qotd-ratings-
5444ddd79c-gp7kb","uid":"d9b9549f-0529-11ec-9802-0617a6095287"},
"namespace":"qotd","replicaset":{"name":"qotd-ratings-5444ddd79c"},"labels":{"app":"qotd-
rating","pod-template-hash":"1000888357"},"container":{"name":"qotd-
rating","image":"registry.gitlab.com/quote-of-the-day/qotd-ratings-service:build-
354180024"},"deployment":{"name":"qotd-ratings"},"node":{"hostname":"padme-okd.coc-
ibm.com","name":"padme-okd.coc-ibm.com","uid":"193183f9-01bd-11ec-926b-0617a6095287","labels":
{"beta_kubernetes_io/arch":"amd64","beta_kubernetes_io/os":"linux","kubernetes_io/hostname":"p
adme-okd.coc-ibm.com","node-role_kubernetes_io/compute":"true","node-
role_kubernetes_io/infra":"true","node-role_kubernetes_io/master":"true"}}},"ecs":
{"version":"1.8.0"},"agent":{"name":"padme-okd.coc-ibm.com","type":"filebeat",
"version":"7.12.1","hostname":"padme-okd.coc-ibm.com","ephemeral_id":"39a50623-e967-451f-b30d-
f06fd421d0cf","id":"456adf88-c87d-49d2-8aa2-9b93d39ba057"}}
    ,"sort":[1629846420053,610747]},

    {"_index":"filebeat-7.12.1-2021.08.20-
000001","_type":"_doc","_id":"qppqensBxt31inA6h5Qa","_score":null,"_source":
{"@timestamp":"2021-08-24T23:07:00.578Z","stream":"stdout","message":"incident euid=5896
epid=8870 dsteuid=4250 dstepid=5530 vd=Sunny39 type=undefined subtype=undefined
level=notice","kubernetes":{"pod":{"name":"qotd-ratings-5444ddd79c-gp7kb","uid":"d9b9549f-
0529-11ec-9802-0617a6095287"},"namespace":"qotd","replicaset":{"name":"qotd-ratings-
5444ddd79c"},"labels":{"app":"qotd-rating","pod-template-hash":"1000888357"},"container":
{"image":"registry.gitlab.com/quote-of-the-day/qotd-ratings-service:build-
354180024","name":"qotd-rating"},"deployment":{"name":"qotd-ratings"},"node":{"name":"padme-
okd.coc-ibm.com","uid":"193183f9-01bd-11ec-926b-0617a6095287","labels":
{"beta_kubernetes_io/os":"linux","kubernetes_io/hostname":"padme-okd.coc-ibm.com","node-
role_kubernetes_io/compute":"true","node-role_kubernetes_io/infra":"true","node-
role_kubernetes_io/master":"true","beta_kubernetes_io/arch":"amd64"},"hostname":"padme-
okd.coc-ibm.com"},"namespace_uid":"dcc7d8e8-01bf-11ec-9802-0617a6095287"},"ecs":
{"version":"1.8.0"},"host":{"name":"padme-okd.coc-ibm.com"},"log":{"offset":611253,"file":
{"path":"/var/log/containers/qotd-ratings-5444ddd79c-gp7kb_qotd_qotd-rating-
bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53.log"}},"input":
{"type":"container"},"container":{"image":{"name":"registry.gitlab.com/quote-of-the-day/qotd-
ratings-service:build-
354180024"},"runtime":"docker","id":"bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389
f5ffe53"},"agent":{"id":"456adf88-c87d-49d2-8aa2-9b93d39ba057","name":"padme-okd.coc-
ibm.com","type":"filebeat","version":"7.12.1","hostname":"padme-okd.coc-
ibm.com","ephemeral_id":"39a50623-e967-451f-b30d-f06fd421d0cf"}},"sort":
[1629846420578,611253]},

    {"_index":"filebeat-7.12.1-2021.08.20-
000001","_type":"_doc","_id":"vJpqensBxt31inA6i5QD","_score":null,"_source":
```

```
{"@timestamp":"2021-08-24T23:07:01.057Z","input":{"type":"container"},"kubernetes":{"labels":
{"app":"qotd-rating","pod-template-hash":"1000888357"},"container":{"name":"qotd-
rating","image":"registry.gitlab.com/quote-of-the-day/qotd-ratings-service:build-
354180024"},"deployment":{"name":"qotd-ratings"},"node":{"uid":"193183f9-01bd-11ec-926b-
0617a6095287","labels":{"node-role_kubernetes_io/compute":"true","node-
role_kubernetes_io/infra":"true","node-
role_kubernetes_io/master":"true","beta_kubernetes_io/arch":"amd64","beta_kubernetes_io/os":"l
inux","kubernetes_io/hostname":"padme-okd.coc-ibm.com"},"hostname":"padme-okd.coc-
ibm.com","name":"padme-okd.coc-ibm.com"},"namespace_uid":"dcc7d8e8-01bf-11ec-9802-
0617a6095287","pod":{"name":"qotd-ratings-5444ddd79c-gp7kb","uid":"d9b9549f-0529-11ec-9802-
0617a6095287"},"namespace":"qotd","replicaset":{"name":"qotd-ratings-
5444ddd79c"}},"container":
{"id":"bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53","image":
{"name":"registry.gitlab.com/quote-of-the-day/qotd-ratings-service:build-
354180024"},"runtime":"docker"},"host":{"name":"padme-okd.coc-ibm.com"},"log":{"file":
{"path":"/var/log/containers/qotd-ratings-5444ddd79c-gp7kb_qotd_qotd-rating-
bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53.log"},"offset":611542},"strea
m":"stdout","message":"incident euid=2210 epid=3789 dsteuid=3713 dstepid=4212 vd=Garrick40
type=undefined subtype=undefined level=notice","agent":{"name":"padme-okd.coc-
ibm.com","type":"filebeat","version":"7.12.1","hostname":"padme-okd.coc-
ibm.com","ephemeral_id":"39a50623-e967-451f-b30d-f06fd421d0cf","id":"456adf88-c87d-49d2-8aa2-
9b93d39ba057"},"ecs":{"version":"1.8.0"}},"sort":[1629846421057,611542]},

    {"_index":"filebeat-7.12.1-2021.08.20-
000001","_type":"_doc","_id":"vZpqensBxt31inA6i5QD","_score":null,"_source":
{"@timestamp":"2021-08-24T23:07:01.584Z","log":{"offset":611726,"file":
{"path":"/var/log/containers/qotd-ratings-5444ddd79c-gp7kb_qotd_qotd-rating-
bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53.log"}},"input":
{"type":"container"},"container":{"image":{"name":"registry.gitlab.com/quote-of-the-day/qotd-
ratings-service:build-
354180024"},"runtime":"docker","id":"bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389
f5ffe53"},"message":"incident euid=8003 epid=5254 dsteuid=2568 dstepid=6719
vd=Sylvan.Romaguera type=undefined subtype=undefined
level=notice","stream":"stdout","kubernetes":{"pod":{"uid":"d9b9549f-0529-11ec-9802-
0617a6095287","name":"qotd-ratings-5444ddd79c-gp7kb"},"namespace":"qotd","replicaset":
{"name":"qotd-ratings-5444ddd79c"},"labels":{"app":"qotd-rating","pod-template-
hash":"1000888357"},"container":{"name":"qotd-rating","image":"registry.gitlab.com/quote-of-
the-day/qotd-ratings-service:build-354180024"},"deployment":{"name":"qotd-ratings"},"node":
{"name":"padme-okd.coc-ibm.com","uid":"193183f9-01bd-11ec-926b-0617a6095287","labels":{"node-
role_kubernetes_io/compute":"true","node-role_kubernetes_io/infra":"true","node-
role_kubernetes_io/master":"true","beta_kubernetes_io/arch":"amd64","beta_kubernetes_io/os":"l
inux","kubernetes_io/hostname":"padme-okd.coc-ibm.com"},"hostname":"padme-okd.coc-
ibm.com"},"namespace_uid":"dcc7d8e8-01bf-11ec-9802-0617a6095287"},"host":{"name":"padme-
okd.coc-ibm.com"},"agent":{"ephemeral_id":"39a50623-e967-451f-b30d-
f06fd421d0cf","id":"456adf88-c87d-49d2-8aa2-9b93d39ba057","name":"padme-okd.coc-
ibm.com","type":"filebeat","version":"7.12.1","hostname":"padme-okd.coc-ibm.com"},"ecs":
{"version":"1.8.0"}},"sort":[1629846421584,611726]},
    {"_index":"filebeat-7.12.1-2021.08.20-
000001","_type":"_doc","_id":"0ppqensBxt31inA6jpTq","_score":null,"_source":
{"@timestamp":"2021-08-24T23:07:02.092Z","stream":"stdout","input":
{"type":"container"},"kubernetes":{"node":{"name":"padme-okd.coc-ibm.com","uid":"193183f9-
01bd-11ec-926b-0617a6095287","labels":{"node-role_kubernetes_io/infra":"true","node-
role_kubernetes_io/master":"true","beta_kubernetes_io/arch":"amd64","beta_kubernetes_io/os":"l
inux","kubernetes_io/hostname":"padme-okd.coc-ibm.com","node-
role_kubernetes_io/compute":"true"},"hostname":"padme-okd.coc-
ibm.com"},"namespace_uid":"dcc7d8e8-01bf-11ec-9802-0617a6095287","pod":{"uid":"d9b9549f-0529-
11ec-9802-0617a6095287","name":"qotd-ratings-5444ddd79c-
gp7kb"},"namespace":"qotd","replicaset":{"name":"qotd-ratings-5444ddd79c"},"labels":{"pod-
template-hash":"1000888357","app":"qotd-rating"},"container":{"name":"qotd-
rating","image":"registry.gitlab.com/quote-of-the-day/qotd-ratings-service:build-
354180024"},"deployment":{"name":"qotd-ratings"}},"ecs":{"version":"1.8.0"},"agent":
{"version":"7.12.1","hostname":"padme-okd.coc-ibm.com","ephemeral_id":"39a50623-e967-451f-
b30d-f06fd421d0cf","id":"456adf88-c87d-49d2-8aa2-9b93d39ba057","name":"padme-okd.coc-
ibm.com","type":"filebeat"},"log":{"file":{"path":"/var/log/containers/qotd-ratings-
5444ddd79c-gp7kb_qotd_qotd-rating-
bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53.log"},"offset":612021},"messa
ge":"incident euid=2850 epid=7575 dsteuid=3306 dstepid=2473 vd=Keshaun_Berge96 type=undefined
subtype=undefined level=notice","container":
{"id":"bc5dd98411a160669c3b0cf8af424f51d0a504cc27f7026a682318389f5ffe53","image":
{"name":"registry.gitlab.com/quote-of-the-day/qotd-ratings-service:build-
354180024"},"runtime":"docker"},"host":{"name":"padme-okd.coc-ibm.com"}},"sort":
```

```
[1629846422092,612021]}
]
```


Log anomaly alert example

This example alert incorporates data from the identified log anomalies into an ObjectServer alert structure. Note that based on the values in the **details.log_anomaly_detector.causality.template_list** field, three different templates have been identified as occurring with significant deviations from the model and have been grouped together within this anomaly.


```
[
    {"id": "a21bd7c9-b3f6-3684-b900-12057634cb66",
    "occurrenceTime": "2021-08-24T23:07:00.000Z",
    "sender": {"name": "Log Anomaly"},
    "event_link": "https://api.us-south.logging.cloud.ibm.com/v1/export?
to=1629846430000&from=1629846420000",
    "summary": "Abnormal behavior found in component: qotd-rating",
    "description": "[<NUM>] Ratings request, id: <NUM>: 3\n[<NUM>] The monkey's dart hit the
<NUM>: 4\nHeartbeat status: CPU=\"OK\" Memory=\"OK\": 2\nincident euid=<NUM> epid=<NUM>
dsteuid=<NUM> dstepid=<NUM> <*>type=undefined subtype=undefined level=notice: 1",
    "severity": 4,
    "type":
        {"eventType": "problem",
        "classification": "log anomaly",
        "condition": "0d33c050-101f-11ec-881b-acde48001122"},
    "resource":
        {"name": "qotd-rating",
        "type": "application",
        "application": "qotd-rating",
        "entity": "qotd-rating"},
    "expirySeconds": 900,
    "details":
        {"log_anomaly_detector":
          {"start_timestamp": 1629846420000,
          "end_timestamp": 1629846430000,
          "original_group_id": "0d33c050-101f-11ec-881b-acde48001122",
          "causality":
            {"service": ["qotd-rating"]},
            "detected_at": 1632252299670.0,
            "source_application_id": "qotd-rating",
            "log_anomaly_confidence": 0.9152,
            "log_anomaly_model": ["PCA"],
            "prediction_error":
              {"pca_prediction_error": 26.884115203263626,
              "embedding_prediction_error": 0.02826894412475822},
            "error_templates": [],
            "template_list": ["Heartbeat status: <*><*><*>", "Heartbeat status: <*><*><*><*>
<*>", "incident euid=<NUM> epid=<NUM> dsteuid=<NUM> dstepid=<NUM> <*>type=undefined
subtype=undefined level=notice", "Heartbeat status: CPU=\"OK\" Memory=\"OK\"", "[<NUM>]
Ratings request, id: <NUM>", "[<NUM>] That doesn't seem fair.", "[<NUM>] The monkey's dart hit
the <NUM>", "Heartbeat status: <*><*><*><*><*><*><*>", "Heartbeat status: <*><*><*><*>",
"Unknown_normal", "Unknown_error"],
            "count_vector": [0, 0, 21, 0, 5, 0, 5, 2, 0, 0, 0],
            "text_dict":
              {"template_ids": ["7eddc431", "b59d9df4", "7cd5f6b4"],
              "template_list": ["Heartbeat status: CPU=\"OK\" Memory=\"OK\"", "Heartbeat
status: <*><*><*><*><*><*><*>", "incident euid=<NUM> epid=<NUM> dsteuid=<NUM> dstepid=<NUM>
<*>type=undefined subtype=undefined level=notice"],
              "count_vector": [0, 2, 21],
              "expected_count_vector": [14.24, 14.91, 15.13],
              "expected_count_lower_bound": [5.34, 6.45, 10.46],
              "expected_count_upper_bound": [23.14, 23.36, 20.02]},
              "application_group_id": "1000",
              "application_id": "1000",
              "model_version": "v4",
              "severity_from_model": 0,
              "description": "Heartbeat status: <*><*><*><*><*><*>: 2\nincident euid=<NUM>
epid=<NUM> dsteuid=<NUM> dstepid=<NUM> <*>type=undefined subtype=undefined level=notice:
21\n[<NUM>] Ratings request, id: <NUM>: 5\n[<NUM>] The monkey's dart hit the <NUM>: 5"}},
              "timestamp": 1632252299670.0}
  ]
```

About log anomaly detection - statistical baseline

Statistical baseline log anomaly detection is an unsupervised learning algorithm that uses a statistical moving average on all of your log data to discover abnormal behavior. This algorithm automatically detects unusual patterns in logs and notifies you when they occur.

Data that is used for analysis is updated every 30 minutes, so this algorithm provides value quickly. After it takes the first 30 minutes to establish a baseline, statistical baseline log anomaly detection uses the computed baseline to identify potential errors.

For domain-specific log anomaly detection, such as for IBM MQ or WebSphere, the statistical baseline model already has a baseline for domain-specific logs. Thus, the model can detect potential errors without waiting for 30 minutes to establish an initial baseline. Other than the means of establishing a baseline, domain-specific log anomaly detection has identical behavior to standard statistical baseline log anomaly detection. For more information about domain-specific log anomaly detection, see this video.

The algorithm for both statistical baseline log anomaly detection and domain-specific log anomaly detection supplements the natural language log anomaly detection algorithm by reporting extra log anomalies to your ChatOps interface and to the Alert Viewer.

Domain-specific log anomaly detection is available for the following products:

- WebSphere
- IBM MQ

Enabling this algorithm helps IBM Cloud Pak for AIOps understand normal behavior so when an anomalous situation arises, you are notified. Anomalies are raised to your ChatOps interface. For information on an anomaly, you can view related alerts to see the patterns and associated log messages.

By default the training of this AI algorithm is enabled and it is suggested that you leave it on. If you disable it, you won't see details about statistical baseline anomalies in either your ChatOps feed or in the Alert Viewer.

For more information about natural language log anomaly detection, see the following topics:

- Workflow
- Examples

Workflow of statistical baseline log anomaly detection

Find out how statistical baseline log anomaly detection processes your log data to generate log alerts.

This algorithm extracts specific entities from the logs, such as short text strings that indicate error codes and exceptions. Combined with other statistics, such as the number of logs for each component, the algorithm uses these statistics as a baseline to discover abnormal behavior in your live log data.

These extracted entities are not just a simple dictionary, but rather use *clue words* to interpret other data in the logs, such as numbers. For example, the number **`500`** appearing in a log message might be a count, identifier, or some other value, but if it is accompanied by clue words, such as **`http status`**, then the system knows that this number refers to an error code. When enough of these deviations are detected to be statistically significant, then the system knows it needs to report an anomaly issue.

Domain-specific log anomaly detection, such as that for IBM MQ or WebSphere, uses the same algorithm as statistical baseline log anomaly detection. However, the entities that are used in domain-specific log anomaly detection and that are included in the input logs are already provided from prior knowledge about the specific domain. After entities like the message ID and log level are detected, the workflow of a domain-specific log anomaly is similar to the workflow of a general statistical baseline log anomaly.

The following figure shows a simplified workflow of the statistical baseline log anomaly detection algorithm.

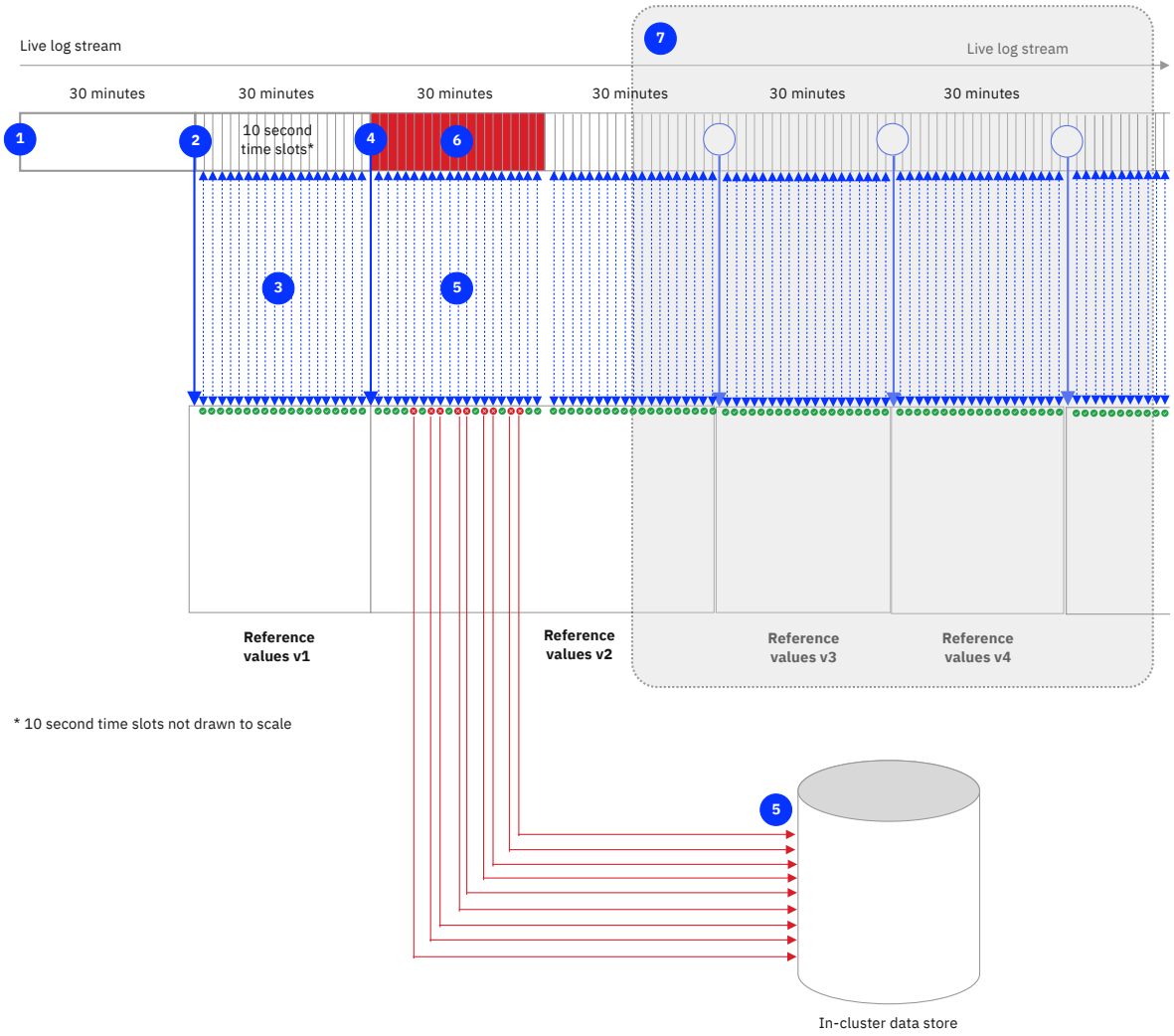


Figure. Workflow

1. **Algorithm switched on**

   The algorithm is first switched on. **Note:** See Supported resource number and throughput rates for limits on throughput rates.

2. **Extract initial reference values**

   At the end of the first 30-minute time interval, the algorithm reviews the log messages from the previous 30 minutes and extracts text entities from the logs. It saves the number of occurrences of each entity as reference values.

3. **Review live stream for anomalies**

   Now that it has the reference values, the system starts monitoring the live log stream for anomalies, which are statistically significant deviations from the baseline. It evaluates each component's logs every 10 seconds for anomalies.

4. **Extract values from next 30-minute interval and update reference values**

   At the end of the next 30-minute time interval, the algorithm again reviews the log messages from the previous 30 minutes. As before, the algorithm extracts text entities and statistics from the logs. It then combines this data with the previous baseline to compute a new baseline, which now covers 60 minutes' worth of normal behavior data.

   Note: If the live data that is being processed is not from the last 30 minutes, then the algorithm still looks for anomalies, but the reference values are not updated. One reason why the last 30-minute time interval might contain older data is when a delay in data processing occurs due to a temporary malfunction of a data processing component, such as a Flink job.

5. **Log alerts are generated**

   The system is continuously monitoring the live log stream for anomalies and evaluates each component's logs every 10 seconds for anomalies. Log anomalies are generated at the end of each 10-second interval for each log line in which a statistical difference exists between the number of entities that are identified and the reference values and numbers, or both, that are identified and accompanied by clue words.

   Based on this analysis, the log anomaly detection algorithm identifies one or more of the following log anomaly types:

   - An entity is detected but not expected and includes an error.
   - An entity is detected with lower frequency than expected.
   - An entity is detected with higher frequency than expected.

   The algorithm creates a log alert for each log anomaly and sends these alerts to the in-cluster data store. Any alerts that are generated by this algorithm are assigned severity 5 (Major).

6. **Interval is potentially tagged as biased**

   Based on the number of log anomalies detected in the previous step, a threshold calculation is performed. The number of log anomalies that are detected is divided by the number of windows. If the resulting value exceeds the predefined threshold of 0.15, then the entire interval is tagged as biased, and none of the reference values are updated based on the entity extraction values for this interval.

   Note: The predefined threshold ensures that frequently occurring anomalies do not bias the statistical baseline reference. If the anomalies are not frequently occurring, such as when the resulting value does not exceed the predefined threshold of 0.15, the interval is not tagged as biased, and the reference values are updated. It is possible that such infrequent anomalies are learned as normal behavior, and they might not be detected again. To reset the learning, you need to manually reset the reference of the statistical baseline. For more information, see Resetting the reference of the statistical baseline.

7. **Processing of the live log stream continues**

   The algorithm continues processing 30-minute time intervals in the live log stream.

Examples of statistical baseline log anomaly detection

This topic provides examples of log anomalies generated by the statistical baseline log anomaly detection algorithm and when they are likely to be found.

Examples presented in this topic

The following examples are described in this topic.

- Unexpected 500 error
- WebSphere log anomaly
- IBM MQ log anomaly

Example: unexpected 500 error

You recently installed the sock-shop application, which is made up of multiple microservices, such as **`front-end`** and **`orders`**. Most of the microservices that make up this application generate their own log files.

Note For the statistical baseline log anomaly detection algorithm to work, there is no formal minimum log line count. The main requirement is that all microservices generate a reasonable amount of log data.

The example proceeds as follows:

1. Initially the microservices are all running OK and generate clean log files, without any text entities that might indicate an error. See Clean log file lines for an example of what a log like this might look like.

2. Shortly after this, some of the pods associated with the microservices start shutting down unexpectedly; as a result of this, **`HTTP 500`** errors appear for the first time in the live log stream.

3. Live log data is analyzed in 10 second slots. Now, when the latest logs are compared to the baseline reference values, several error entities are identified: **`500`**, **`error`**, and **`illegIllegalStateException`**. The algorithm determines that the number **`500`** is an error, as it is accompanied by the entity **`status`**, which points to an HTTP 500 error, a general HTTP status code indicating that something has gone wrong on a server but the server could not be more specific about what the exact problem is. These entities have not been identified before during normal behavior and since they are statistically significant (trivially so in this example) thus point to an anomaly. See Error log file lines for an example of what a log like this might look like.

   Note If this was an error message that occurs on a regular basis, say an optional library that is not used in a given piece of code, then the statistical baseline log anomaly detection algorithm will automatically learn this, and consequently treat this error message as normal and therefore no longer alert on it.

4. Log anomaly alerts are generated for each log anomaly and sent to the in-cluster ObjectServer. See Log anomaly alert format for an example of what a log anomaly alert might look like.

Reference information

The following information is referenced in this task.

- Clean log file lines
- Error log file lines
- Log anomaly alert example

Clean log file lines

These example JSON log snippets display clean log lines for the front-end and orders microservices within the sock-shop application. The log algorithm extracts entities from the **`_line`** keyword within these log lines and then performs the analysis described in Workflow of statistical baseline log anomaly detection on these extracted entities.

Clean log file lines from the front-end microservice

The following log files lines do contain numbers but they are not accompanied by error entities. The numbers in these log message are actually part of user address data (for example **`322 Main Street`**) and credit card data, and can be seen in the **`_line`** parameter in the code snippet below. You might need to scroll to the right to see the full content of the **`_line`** parameter.

```
[
  {
    "_app":"front-end",
    "_file":"/var/log/containers/front-end-85c54f4fb-jvzqn_sock-shop_front-end-
e19f0a886c8a03aa63e8138def1df8f33a60981ec41cdbc2aca7c120d16e7a42.log",
    "_host":"front-end-85c54f4fb-jvzqn",
    "_ip":"172.30.148.208",
    "_ipremote":"52.117.50.10",
    "_lid":"1264263893737455616",
    "_line":"Posting Address: {\"number\":\"\",\"street\":\"322 Main
Street\",\"city\":\"Gotham City\",\"postcode\":\"12345\",\"country\":\"United
States\",\"userID\":\"1q2w3e4r5t6y7u8i9o0p\"}",
    "_logtype":"json",
    "_mac":"00:00:00:00:94:60",
    "_tag":["k8s"],"_ts":1601424000273,
    "city":"Gotham City",
    "container":"front-end",
    "containerid":"e19f0a886c8a03aa63e8138def1df8f33a60981ec41cdbc2aca7c120d16e7a42",
    "country":"United States",
    "namespace":"sock-shop",
    "node":"kube-bqvhtmdd031d903ud4g0-simulationd-default-00000105",
    "number":"",
    "pod":"front-end-85c54f4fb-jvzqn",
    "postcode":"12345",
    "street":"322 Main Street",
    "userID":"0p9o8i7u6y5t4r3e2w1q"
  },
  {
    "_app":"front-end",
    "_file":"/var/log/containers/front-end-85c54f4fb-jvzqn_sock-shop_front-end-
e19f0a886c8a03aa63e8138def1df8f33a60981ec41cdbc2aca7c120d16e7a42.log",
```

```
    "_host":"front-end-85c54f4fb-jvzqn",
    "_ip":"172.30.148.208",
    "_ipremote":"52.117.50.10",
    "_lid":"1264263893737455617",
    "_line":"\u001B[0mPOST /addresses \u001B[32m200 \u001B[0m9.663 ms - -\u001B[0m",
    "_logtype":"customapp",
    "_mac":"00:00:00:00:94:60",
    "_tag":["k8s"],
    "_ts":1601424000282,
    "container":"front-end",
    "containerid":"e19f0a886c8a03aa63e8138def1df8f33a60981ec41cdbc2aca7c120d16e7a42",
    "namespace":"sock-shop",
    "node":"kube-bqvhtmdd031d903ud4g0-simulationd-default-00000105",
    "pod":"front-end-85c54f4fb-jvzqn"
  }
]
```


Clean log file lines from the orders microservice

The following log files lines do contain numbers but they are not accompanied by error entities. The numbers in these log message are actually part of user address data (for example **`322 Main Street`**) and credit card data, and can be seen in the **`_line`** parameter in the code snippet below. You might need to scroll to the right to see the full content of the **`_line`** parameter.


```
[
    {"_app":"orders",
    "_file":"/var/log/containers/orders-b8cb69f5c-gn87g_sock-shop_orders-
9d699fcc72f3283bf66ecb0bb850251e73dd5a7f5a0026b4fcb6facc6c533baf.log",
    "_host":"orders-b8cb69f5c-gn87g",
    "_ip":"172.30.148.208",
    "_ipremote":"52.117.50.10",
    "_lid":"1264263925219901440",
    "_line":"2020-09-30 00:00:07.959  INFO [orders,4ecc59a2bbf00a18,276a9b5feb58dcd3,true] 7 -
-- [p-nio-80-exec-8] w.w.s.o.controllers.OrdersController     : Sending payment request:
PaymentRequest{address=Address{id=null, number='', street='322 Main Street', city='Gotahm
City', country='United States', postcode='12345'}, card=Card{id=null,
longNum='1234567812345678', expires='12/56', ccv='321'}, customer=Customer{id=null,
firstName='John', lastName='Doe', username='John_Doe', addresses=[], cards=[]}}",
    "_logtype":"json",
    "_mac":"00:00:00:00:94:60",
    "_tag":["k8s"],
    "_ts":1601424007959,
    "addresses":"[],",
    "cards":"[]}}",
    "container":"orders",
    "containerid":"9d699fcc72f3283bf66ecb0bb850251e73dd5a7f5a0026b4fcb6facc6c533baf",
    "firstName":"'John',",
    "lastName":"'Doe',",
    "level":"INFO",
    "message":"[orders,4ecc59a2bbf00a18,276a9b5feb58dcd3,true] 7 --- [p-nio-80-exec-8]
w.w.s.o.controllers.OrdersController     : Sending payment request:
PaymentRequest{address=Address{id=null, number='', street='322 Main Street', city='Gotham
City', country='United States', postcode='12345'}, card=Card{id=null,
longNum='1234567812345678', expires='12/56', ccv='321'}, customer=Customer{id=null,
firstName='John', lastName='Doe', username='John_Doe', addresses=[], cards=[]}}",
    "namespace":"sock-shop",
    "node":"kube-bqvhtmdd031d903ud4g0-simulationd-default-00000105",
    "pod":"orders-b8cb69f5c-gn87g",
    "username":"'John_Doe',"}
]
```


Error log file lines

The following example JSON log snippet displays error log lines for the front-end microservice within the sock-shop application. The log algorithm extracts entities from the **`_line`** keyword within these log lines and then performs the analysis described in Workflow of statistical baseline log anomaly detection on these extracted entities.

Error log file line from the front-end microservice

The following log line contains numbers accompanied by error entities; in particular, the **_line** parameter contains the number **500**, as well as the error entities **status**, **exception**, and **IllegalStateException**. You might need to scroll to the right to see the full content of the **_line** parameter.


```
[
    {"_account": "6a8e297e13",
    "_cluster": "ld70",
    "_host": "front-end-85c54f4fb-jvzqn",
    "_ingester": "logdna-agent/1.6.2 (Ubuntu/16.04) k8s",
    "_logtype": "json",
    "_mac": "00:00:00:00:94:60",
    "_tag": ["k8s"],
    "_file": "/var/log/containers/front-end-85c54f4fb-jvzqn_sock-shop_front-end-
e19f0a886c8a03aa63e8138def1df8f33a60981ec41cdbc2aca7c120d16e7a42.log",
    "_line": "Order response: {\"timestamp\":1599236098757,\"status\":500,\"error\":\"Internal
Server Error\",\"exception\":\"java.lang.IllegalStateException\",\"message\":\"Unable to
create order due to unspecified IO error.\",\"path\":\"/orders\"}",
    "_ts": 1599236098759,
    "_app": "front-end",
    "pod": "front-end-85c54f4fb-jvzqn",
    "namespace": "sock-shop",
    "container": "front-end",
    "containerid": "e19f0a886c8a03aa63e8138def1df8f33a60981ec41cdbc2aca7c120d16e7a42",
    "node": "kube-bqvhtmdd031d903ud4g0-simulationd-default-00000105",
    "_ip": "172.30.148.208",
    "_ipremote": "52.117.50.10",
    "__key": "logline:6a8e297e13:front-end:front-end-85c54f4fb-jvzqn:ld70",
    "_bid": "e4975ea4-6634-4f88-accc-8eba410d2617:61301:ld70",
    "level": null,
    "timestamp": 1599236098757,
    "error": "Internal Server Error",
    "exception": "java.lang.IllegalStateException",
    "message": "Unable to create order due to unspecified IO error.",
    "_status": 500,
    "_path": "/orders",
    "_id": "1255087168836907020"}
]
```


Log anomaly alert example

This example alert incorporates data from the identified log anomalies into an ObjectServer alert structure.

- This alert has been assigned a severity of 5.
- The **summary** parameter explains that 24 errors have been found in the front-end component.
- The **details.log_anomaly_detector.text_dict.template_list** parameter lists the error entities identified.


```
[
    {"id": "d708346f-046a-35ae-b0ab-1a9ad8f1bafa",
    "occurrenceTime": "2020-09-04T16:15:40.000Z",
    "sender": {"name": "Log Anomaly"},
    "event_link": "https://api.us-south.logging.cloud.ibm.com/v1/export?
to=1599236150000&from=1599236140000",
    "summary": "24 errors found in component: front-end",
    "description": "500: 8\njava.lang.IllegalStateException: 8\nerror_log: 8",
    "severity": 5,
    "type":
        {"eventType": "problem",
        "classification": "log anomaly",
        "condition": "c9680808-14bb-11ec-b72f-acde48001122"},
    "resource":
        {"name": "front-end",
        "type": "application",
        "application": "front-end",
        "entity": "front-end"},
    "expirySeconds": 900,
    "details":
        {"log_anomaly_detector":
            {"start_timestamp": 1599236140000,
            "end_timestamp": 1599236150000,
            "original_group_id": "c9680808-14bb-11ec-b72f-acde48001122",
            "causality":
```

```
                {"service": ["front-end"]},
            "detected_at": 1632338288103.0,
            "source_application_id": "front-end",
            "log_anomaly_confidence": 1.0,
            "log_anomaly_model": ["oob_entities"],
            "prediction_error":
                {"oob_entities_error": 24},
            "error_templates": [0, 1, 2],
            "template_list": ["500", "java.lang.IllegalStateException", "error_log"],
            "count_vector": [8, 8, 8],
            "text_dict":
                {"template_ids": ["oob_entities_500",
"oob_entities_java.lang.IllegalStateException", "oob_entities_error_log"],
                "template_list": ["500", "java.lang.IllegalStateException", "error_log"],
                "count_vector": [8, 8, 8],
                "expected_count_vector": [0.29, 0.29, 0.48],
                "expected_count_lower_bound": [0, 0, 0],
                "expected_count_upper_bound": [3.0, 3.0, 3.21]
                },
                "application_group_id": "ss",
                "application_id": "1",
                "model_version": "5",
                "severity_from_model": 0,
                "description": "500: 8\njava.lang.IllegalStateException: 8\nerror_log: 8"
            }
        },
    "timestamp": 1632338288103.0}
  ]
```

Example: WebSphere log anomaly

The following example shows a WebSphere log. In log anomalies for a specific domain like WebSphere, entities are included in the logs. WebSphere log anomaly detection identifies which entities are error indicators. Message ID, **ibm_messageId**, and log levels, **loglevel**, are examples of WebSphere entities.

```
{
  "type": "was_message",
  "host": "abc-123.myserver.com",
  "ibm_cellName": "gsV900Cell",
  "ibm_nodeName": "node4",
  "ibm_serverName": "nodeagent",
  "ibm_sequence": "1616166373781_0000000000002",
  "message": "TRAS0017I: The startup trace state is *=info.",
  "ibm_datetime": "2021-03-19T11:06:13.781-0400",
  "ibm_messageId": "TRAS0017I",
  "ibm_threadId": "00000001",
  "module": "com.ibm.ejs.ras.ManagerAdmin",
  "loglevel": "INFO",
  "unix_ts": 1616166373781,
  "start": 1616166370000,
  "end": 1616166380000
}
```

On the **Incidents and alerts** page, the following example from the **Summary** parameter explains that two logs were found in the **qotd-qrcode** container. This parameter also says that a primary message code of **SSLC0008E** is contributing to the error. This alert is further explained by the **Alert details** panel. For more information, see Domain-specific log anomaly alert details.

```
Log Anomaly found 2 logs containing errors for container(s): qotd-qrcode with a primary
message code of SSLC0008E
```

Example: IBM MQ log anomaly

The following example shows an IBM MQ log. In log anomalies for a specific domain like IBM MQ, entities are included in the logs. IBM MQ log anomaly detection identifies which entities are error indicators. Message ID, **ibm_messageId**, and log levels, **loglevel**, are examples of IBM MQ entities.

```
{
  "ibm_messageId": "AMQ6287I",
  "ibm_arithInsert1": 0,
  "ibm_arithInsert2": 0,
  "ibm_commentInsert1": "Linux 4.18.0-193.70.1.el8_2.x86_64 (Cloud Pak Linux (x86-64 platform)
```

```
64-bit)",
  "ibm_commentInsert2": "/opt/mqm (Installation1)",
  "ibm_commentInsert3": "9.3.0.0 (p930-L220526)",
  "ibm_datetime": "2022-06-01T01:13:49.673Z",
  "ibm_serverName": "QMA",
  "type": "mq_log",
  "host": "slot11-qma-ibm-mq-0",
  "loglevel": "INFO",
  "module": "amqxeida.c:6755",
  "ibm_sequence": "1654046029_673594295",
  "ibm_processId": "37",
  "ibm_threadId": "3",
  "ibm_version": "9.3.0.0",
  "ibm_processName": "amqzmuc0",
  "ibm_userName": "mqm",
  "ibm_installationName": "Installation1",
  "ibm_installationDir": "/opt/mqm",
  "message": "AMQ6287I: IBM MQ V9.3.0.0 (p930-L220526)."
}
```


On the **Incidents and alerts** page, the following example from the **Summary** parameter explains that two logs were found in the `qotd-qrcode` container. This parameter also says that a primary message code of `SSLC0008E` is contributing to the error. This alert is further explained by the **Alert details** panel. For more information, see Domain-specific log anomaly alert details.

```
Log Anomaly found 2 logs containing errors for container(s): qotd-qrcode with a primary
message code of SSLC0008E
```

About metric anomaly detection

This AI algorithm generates alerts when it detects anomalous behavior in your metrics.

Description

Metric anomaly detection is an unsupervised learning algorithm that learns normal patterns of metric behavior by analyzing metric values at regular intervals and raises anomalies or alerts when that behavior significantly changes.

Training this AI model helps IBM Cloud Pak® for AIOps discover metric anomalies. You are then alerted to problems before services or applications are impacted.

Metric anomalies are generated based on the following analysis:

- Dynamic calculation of baseline values for each metric. For example, a metric baseline might be within the 1 - 25 range. If a metric value is returned outside of this range, then a metric anomaly alert is immediately generated.
- Flatlining, whereby the system identifies that the metric started unexpectedly returning a constant value.
- Finite Domain, whereby the system detects an anomaly when a metric value elevates to a level not reached previously.
- Predominant Range, whereby the system detects an anomaly when the variation in a metric value exceeds the range within which the metric normally varies.
- A metric that is identified as varying significantly and is later found to vary less significantly is also flagged as a metric anomaly alert.

Site reliability engineers (SREs) and other users responsible for application and service availability are able to display metric anomalies as alerts within the context of an incident, as described in the following topics:

- Viewing alerts in ChatOps
- Viewing probable cause alerts in an incident

They can also view metric anomaly alerts in the Alert Viewer and associated graphs that display metrics over time, baseline values of metrics, highlighting of anomalous metrics, and forecasting of metric values, as described in the following topics:

- Viewing metric anomaly details
- Metric anomaly forecasts

Note: Individual anomolous or non-anomalous metrics can be viewed on a timeseries chart using the Metric search function located on the Automation console.

Prerequisites

To train metric anomaly detection, you need time series data that takes measurements at regular intervals to detect anomalies.

Metric anomaly detection requires a minimum of 14 days of historical data to help the algorithm understand what a weekend looks like compared to a weekday.

Data ingestion

Data comes from one of the following sources

- Ingest metric data by posting data using the REST API. For more information, see Metrics API and the POST entry in the associated Swagger document.
- Check that ingested data is being ingested by retrieving metric data using the REST API. For more information, see Metrics API and the GET entry in the associated Swagger document.
- If you have a metric data connection set up (Instana, Dynatrace, New Relic, Zabbix, for example), you can ingest data using the relevant connection, as described in Connections.

Automating metric anomalies to create incidents

To automatically create incidents out of particular anomalies, you need to create a policy - or policies - and add the relevant conditions. Here is an example of such a policy:

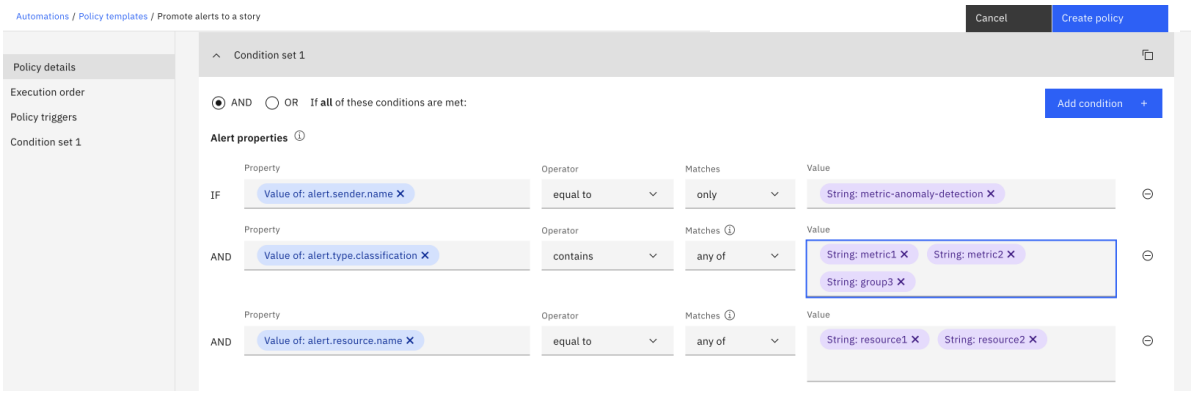


Figure. Policy template alerts

This policy means that the next time there is a new alert on **`resource1`** or **`resource2`** for **`metric1`** or **`metric2`**–or any metric in group3–then an incident is automatically created for it.

For more information, see Promote alerts to an incident.

Language support

For information about supported languages for this algorithm, see Language support.

About probable cause ranking

This AI algorithm identifies the alert with the greatest probability of being the cause of an incident.

It is enabled by default, so that information about the probable cause of a problem is automatically included in your incident details.

Description

Probable cause is a set of approaches and customizations that analyses alerts and topologies together. It then uses that understanding to try to determine the root cause of a problem. After the probable cause alert is identified, action can be taken on the resource that is generating that alert to quickly resolve the incident.

A number of techniques are used to make a probable cause determination. These are derived from topological relationships and natural language calculations. The natural language model has been built to classify the alerts in the golden signal classes, based on their summary. This information is then combined with the topological information (path calculation between the affected resources on which the alert exists). Then, the path and class are combined together to determine the rank of the alert, within the context of the other alerts in the incident.

- Keyword in the **Summary**: For example, if the word "down" appears, it might be more indicative of the root cause than the word "up".
- Severity: The highest severity normally means it's more likely to be indicative of what to look for in the root cause.
- First to occur: Set off by default. Often the first alert to arrive in a group of alerts can be seen as the root cause of the problem, but it is not always the case.
- The ability to inherit previous root causes from IBM Tivoli Network Manager (ITMN) root causes and CauseWeights that exist from scope-based alert grouping in Netcool. (For more information, see the **Event Grouping** section here.)
- Scoring config: The following options contribute to probable cause determination:
  - `PathCalculationEnabled`: This option uses topological data to calculate the path between sources. You must create the topological resource group. For the resource group, set the alert correlation to `true`. This setting allows alerts to be correlated by toplogical correlation. When a problem is encountered in the system, a path is calculated between all the alerts in the group. This path is the shortest number of hops. After the path is calculated, a path score is calculated based of the edge types in the system.
  - `wordEnabled`: You can add or remove words from the system with an API. If a specified word is in the summary field of an alert, the word score is added.
    - Another option exists for case sensitivity.
    - The higher the score, the more likely a problem is to be the root cause.
    - The word score is the sum of the score. For example, if 2 words in the summary are found, and one word has a value of 100, and the other has a value of 10, the total word score is 110.
  - `SeverityEnabled`: The problem severity is accounted for. By default, the score is the severity multiplied by 10.
  - `causeScoreEnabled`: If the field `CauseWeight` is present in the details section of an alert then the `causeScore` is added to the total score in `CauseWeight`. If this field is not present in the details section, there is no effect. This field cannot be changed.
  - `itnmproccessing`: If this option is turned on and ITMN marked an alert as the root cause(`NmosCauseType=1`), then `itnmcauseweight` is added to the score.
  - `firstBoost`: If `firstBoost` is set to true, the amount that is specified in the `firstBoostWeight` is added to the alert that has the lowest `firstoccurrence` time.

The techniques work together to determine the root cause. One technique alone does not work in an environment as an alert might, or might not, have an associated topology.

Prerequisites

For topology and natural language probable cause, the alert data must have topological data and must also be topologically co-related. For more information, see Topological grouping.

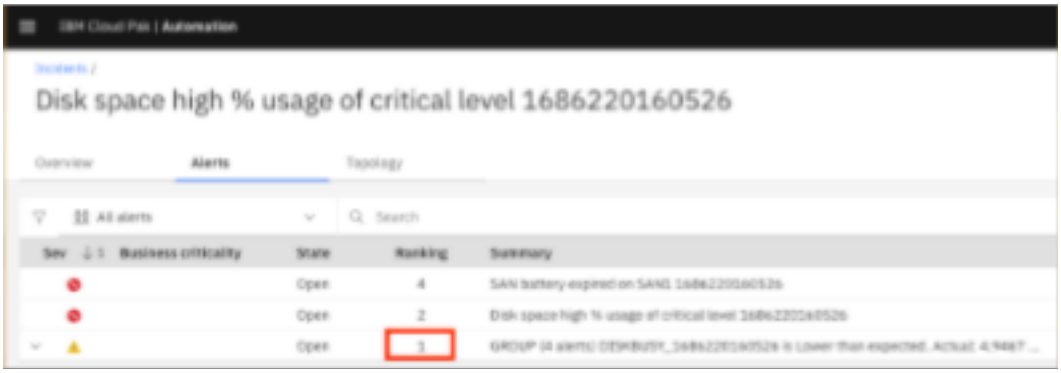
Viewing rankings

Probable cause rankings from 1 (highest) to 3 (lowest) in order of probability can be viewed in the Incident overview UI.



Figure. Probable cause Incidents Overview

They also display (showing rankings greater than 3) in the Alerts tab of the Incidents viewer.



Probable cause Alerts tab

Rankings can also appear in ChatOps notifications. (For more information, see Probable cause in ChatOps.)

Language support

For information about supported languages for this algorithm, see Language support.

Related topic

- Configuring probable cause

About similar tickets

This AI algorithm discovers details about similar tickets that occurred in the past and are impacting the current application.

Description

Important: Starting with IBM Cloud Pak® for AIOps version 4.1.0, similar incidents, such as from a ServiceNow integration, are called similar tickets.

When an incident occurs, it can be helpful to review details for similar tickets from the past to help determine a resolution. Similar tickets is an unsupervised learning algorithm that aggregates information from past similar tickets. It can also extract

the steps used to fix previous tickets, if documented.

Training this AI algorithm will help you discover historical tickets to aid in the remediation of current problems. To train this algorithm, you must complete some initial setup tasks, as described in Setting up training for similar tickets.

Site reliability engineers (SREs) and other users responsible for application and service availability are able to access tickets in their ChatOps interface and within each ticket view historical similar tickets, as described in Managing incidents.

Prerequisites

The Similar tickets algorithm requires ticket data from ServiceNow. Before attempting to set up training for this algorithm, ensure that you have created a functioning data connection to ServiceNow. For more information, see Creating a ServiceNow connection.

After training is complete, enable the similar ticket policy in the **Policies** page. For more information, see Setting up training for similar tickets.

Language support

For information about supported languages for this algorithm, see Language support.

Language support

The AI algorithms in the AI Model Management are able to analyze and train on a limited set of non-English language data. In an operations environment, languages often occur in combination. For example, English and Spanish, or English and German. Analysis and training on the following non-English languages data are supported: German, and Spanish.

The use of mixing languages in a feed of records is supported. However, mixing languages in an individual record is not supported. For example, a feed of support tickets can contain a mixture of English and Spanish tickets, but a mixture of languages in a single support ticket is not supported.

If data in an unsupported language is detected during pre-training of any of the algorithms, **Summary** *Step 1 - Precheck data* will contain the warning message **Data needs improvement**.

Click **View recommendations** and, if unsupported langage is a problem, the following instructions appear :

```
We have detected that a portion of this data is in an unsupported language and could impact the quality of this model.

We recommend that you remove any data containing unsupported languages. Then, run training again.
```

Supported language combinations

The following table lists supported language combinations for each of the AI algorithms.

Table. Supported language combinations

Algorithm	Data used	Supported language combinations	Not supported
Alert Grouping	Events	Not language sensitive	N/A
Change risk	Tickets	English and Spanish, English and German, English and Spanish and German	N/A
Log anomaly detection	Logs	English and Spanish, English and German, English and French, English and Italian, English and Korean	N/A
Metric anomaly detection	Metrics	Not language sensitive	N/A

Algorithm	Data used	Supported language combinations	Not supported
Probable cause	Events	English only	If event data in Spanish or German is processed by this algorithm, the ranking of probable cause in an incident might be unreliable or missing.
Similar tickets	Tickets	English and Spanish, English and German, English and Spanish and German, English and French, English and Italian, English and Japanese	N/A

Preparing AI algorithms

Use this information for how to prepare AI algorithms. Some algorithms require setting up training, training, and deploying, such as trainable AI algorithms. Others simply need to be enabled, and others still are enabled by default.

For more information about preparing AI algorithms, see the following topics:

- About algorithm types
- About pre-trained AI algorithms
- Setting up training for trainable AI algorithms

About algorithm types

The AI algorithms can be categorized into two types: ones that analyze your data in real time, and ones that must first be trained to generate an AI model.

The following are the two types of AI algorithms:

- Trainable AI algorithms
- Pre-trained AI algorithms

Trainable AI algorithms

These algorithms require user interaction as they must first be trained to generate an AI model. Models are well-defined computations that are formed as a result of an algorithm that takes some value, or set of values, as input and produces some value, or set of values as output. When deployed, it is your model that acts on the data and derives insights.

You must first set up training for these AI algorithms, and then train that algorithm to generate a model. The following table lists the trainable AI algorithms in alphabetical order, along with a link to the relevant training setup task.

Algorithm	Action
Change risk	Set up training
Metric anomaly detection	Set up training
Natural language log anomaly detection	Set up training
Similar tickets	Set up training
Temporal grouping	Set up training

Pre-trained AI algorithms

These algorithms analyze your data in real time and derive insights without the need to generate a model.

You do not need to set up training for these AI algorithms. You can enable them. In some cases, these algorithms are automatically enabled for you. The following table lists the online algorithms in alphabetical order, along with their default state, and whether you can take action on them to change that state.

Algorithm	Default state	Action
Probable cause	Enabled	None
Scope-based grouping	Enabled	None
Statistical baseline log anomaly detection	Enabled	Disable or Enable
Topological grouping	Enabled	None

About pre-trained AI algorithms

These algorithms don't require models to derive insights. They are enabled by default. You do not need to set up training for these AI algorithms; they are automatically enabled for you, except for the statistical baseline log anomaly detection algorithm, which is enabled by default, but can be disabled.

Click here for more information about the pre-trained AI algorithms:

- About probable cause
- About scope-based grouping
- About statistical baseline log anomaly detection
- About topological grouping

About probable cause

The probable cause AI algorithm is enabled by default and cannot be disabled. To indicate this, the word **Enabled** appears in grayed out text at the bottom of the "Probable cause" tile in the AI Model Management.

For more information on the probable cause AI algorithm, see About probable cause ranking.

About scope-based grouping

The scope-based grouping algorithm is enabled by default and cannot be disabled. To indicate this, the word **Enabled** appears in grayed out text at the bottom of each of the event grouping tiles in the AI Model Management.

For more information on the event grouping AI algorithms, see About scope-based grouping.

About statistical baseline log anomaly detection

Leave this AI algorithm turned on to see details about statistical baseline anomalies in your ChatOps feed and in your Alert Viewer.

Statistical baseline log anomaly detection is an unsupervised learning algorithm that automatically detects unusual patterns in logs and notifies you when they occur. Data that is used for analysis is updated every 30 minutes so this algorithm provides value quickly. This algorithm supplements the natural language log anomaly detection algorithm by reporting extra log anomalies to your ChatOps interface and to the Alert Viewer.

By default, this algorithm is turned on and it is recommended that you leave it turned on. If you turn it off, you won't see details about statistical baseline anomalies in either your ChatOps feed or in your Alert Viewer.

Note: If both statistical baseline and natural language log anomaly detection algorithms are active, then any log anomalies discovered by both will be reconciled, so that only one alert is generated.

About topological grouping

The topological grouping AI algorithm is enabled by default and cannot be disabled. To indicate this, the word **Enabled** appears in grayed out text at the bottom of each of the event grouping tiles in the AI Model Management.

For more information on the event grouping AI algorithms, see About topological grouping.

Setting up training for trainable AI algorithms

AI algorithms analyze different kinds of data and produce AI models, which automate research, reducing the need for manual work in solving problems. IBM Cloud Pak® for AIOps training generates models for a number of different AI algorithms. You can train the models for each of these AI algorithms independently of each other.

For more information about setting up training for trainable AI algorithms, see the following topics:

- Setting up training for change risk
- Setting up training for natural language log anomaly detection
- Setting up training for similar tickets
- Setting up training for temporal grouping
- Setting up training for metric anomaly detection

Setting up training for change risk

Set up training for this AI algorithm to determine how likely it is that a proposed change would cause a problem.

About this task

Hundreds of changes can affect an application during its lifecycle. Change risk is an unsupervised learning algorithm that correlates historical change data with historical incident data and determines how likely it is that a specific code change would cause a problem. Using the assessment score provided, you can determine how safe it is to proceed with the change. Training this AI algorithm helps you ensure that risky changes for an application are assessed before deployment.

Training setup of the algorithm is done by completing some setup tasks. When the training setup is complete, you need to go to a different part of the AI Model Management to train the model.

Before you begin

To train this algorithm, you need change request and incident ticket data from ServiceNow. If you enable the collection of historical data when you connect to ServiceNow, then data for the selected timeframe is available to train on.

To provide the best data for change risk, you need a mix of successful and failed change tickets. Around 10,000 change tickets, with a minimum 2% failure rate, is ideal. However, models can also be created with much less.

Closed change ticket count requirements

For a closed change ticket count less or equal to 150, you need at least three problematic tickets. For a closed change ticket count greater than 150, you need at least 2% of change tickets to be problematic. A *problematic change ticket* is defined as a change ticket that can "either be associated with incident tickets or have an 'unsuccessful' value set in the change ticket close code field".

Failure rate

If you provide data with a change ticket failure rate of less than 2%, the training fails with a **Needs improvement** data quality error. For more information, see Data quality messages.

To check whether you meet these requirements, review your ServiceNow connection. For more information, see Creating a ServiceNow connection.

To resolve the issue, you can optionally configure the failure percentage. Edit the deployment yaml of the ai-platform server and reduce the value for the failure percentage to do so. The following example shows the failure percentage set to 1.9.

```
oc set env deploy/aimanager-aio-ai-platform-api-server MIN_LABEL_PERCENT="1.9"
```

Where **MIN_LABEL_PERCENT** is the failure percentage.

Note Model quality deteriorates with a MIN_LABEL_PERCENT less than 2.0

Running the task

There are multiple parts to this task:

- Starting the training setup
- Checking data connections
- Deciding how to run the training
- Deciding how to deploy the training
- What to do next

Starting the training setup

Go to the AI Model Management and complete some setup tasks.

1. In the Cloud Pak for AIOps home page, click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **AI model management** to open the AI Model Management.
3. Under the Training tab, in the Trainable AI algorithms section, click the **Change risk** tile.

Note: If you want to edit configurations for algorithms after the first time set-up, you can simply click the algorithm tile.

Checking data connections

Review the information on the page. It explains what this AI algorithm is, how it helps in your production environment, and provides a list of data and tool connections that are needed to generate a model.

1. On the Data and tool connections, check that at least one connection is listed.

   Note: If no connections are listed, or you expected to see more connections, click **Go to Data and tool connections** to modify your connections. For more details, see Creating a ServiceNow connection.

2. Click **Next** to move to the next section.

Deciding how to run the training

Decide whether to run the training setup on demand or to schedule it to run on an ongoing basis.

1. Proceed as follows:
   - To run the training setup on demand, ensure that **Schedule to run** is set to **Off**, and go to step 3.
   - To run the training setup on a schedule, set **Schedule to run** to **Yes**, and go to the next step.
2. Schedule the run. Click this option to specify a schedule. You can specify a start date with an optional end date, a frequency, and a time based on Coordinated Universal Time (UTC).
3. Click **Next** to move to the next section.

Deciding how to deploy the training

Decide whether to review the results of training before deploying the model, or to have the system automatically deploy the model.

1. Proceed as follows:
   - To review the results of training before deploying the model, ensure that **Deployment type** is set to **Review first**.
   - For the system to automatically deploy the model, set **Deployment type** to **On completion**.
2. Click **Save** to save the training setup for the algorithm.

What to do next

Now that the training setup is complete, you can train the algorithm. For more details, see Launching the training.

Enabling full data profiling report in Change Risk

Note: To enable this feature, a user must have administrator privilege.

The following procedure generates a JSON data profiling report. This JSON report provides information about word count and parts of speech in the tickets that are used for training, which can be helpful in understanding model behavior.

1. Obtain the change-risk algorithm of the corresponding namespace as a yaml file.

   ```
   oc -n get algorithm change-risk -o yaml >temp.yaml; cp temp.yaml backup.yaml
   ```

2. Use a text editor to extract the value of the **manifestBase64** field from **temp.yaml**, then use **base64** to decode it.

   ```
   echo <value of manifestBase64> | base64 -d > manifest.tmp
   ```

3. Edit **manifest.tmp** to add **GENERATE_FULL_PROFILING_REPORT** to the environment. Set it to **True**.

   ```
   env:
     - name: GENERATE_FULL_PROFILING_REPORT
       value: "True"
   ```

4. Encode the updated file. Update the value of **manifestBase64** in **temp.yaml** with this result.

   ```
   cat manifest.tmp | base64
   ```

5. Apply the modified **temp.yaml** on the cluster.

   ```
   oc apply -f temp.yaml
   ```

6. When enabled, the full data profiling report will be generated from the next run of the change-risk training.

Accessing full data profiling report

The report of the corresponding training is stored in Elasticsearch under the index name **change-risk-data-profiling-<Version-Number>** Access the *version-number* from the AI model management tool when you run training.

Example: if Version number is **v1** then the report is in **change-risk-data-profiling-v1**

To access Elasticsearch, obtain the Elasticsearch username and password from the Elasticsearch secret, and encode it in the request.

```
curl -k --location --request GET 'https://iaf-system-elasticsearch-es-<NAMESPACE>.apps.
<CLUSTER_NAME>.mysite.ibm.com/change-risk-data-profiling-v1/_search?pretty' \
--header 'Authorization: Basic <AUTH>
```

– where AUTH is base64 encoded value of username:password.

Setting up training for metric anomaly detection

Set up training for this AI algorithm to detect anomalous behaviour in your metrics and notify you by generating an alert when they occur.

For more information on the metric anomaly detection AI algorithm, see About metric anomaly detection.

About this task

There are multiple parts to this task:

- Starting the training setup
- Getting started
- Deciding how to run the training
- Deciding how to deploy the training
- What to do next

Starting the training setup

Go to the AI Model Management and complete some setup tasks.

1. In the Cloud Pak for AIOps home page, click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **AI model management** to open the AI Model Management.
3. In the AI algorithms tab, click **Set up training** within the **Metric anomaly detection** tile.

Getting started

Review the information on this page. It explains what this AI algorithm is, how it will help in your production environment, and provides information on the data needed for the algorithm.

1. Review the information on this panel to understand more about metric anomaly detection.
2. Click **Next** to move to the next panel.

Schedule the training

Decide whether to run the training setup on demand or to schedule it to run on an ongoing basis.

1. Proceed as follows:
   - To run the training setup on demand, ensure that **Schedule to run** is set to **Off**, and go to step 3.
   - To run the training setup on a schedule, set **Schedule to run** to **Yes**, and go to the next step.
2. Schedule the run. Click this option to specify a schedule. You can specify a start date with an optional end date, a frequency, and a time based on Coordinated Universal Time (UTC).
3. Click **Next** to move to the next panel.

Deciding how to deploy the training

The option to review training results before deployment is disabled for this algorithm. The metric anomalies generated by training the algorithm will automatically be displayed in the Alert Viewer when training is complete.

Click **Done** to save the training setup for the algorithm.

What to do next

Now that the training setup is complete, you can train the algorithm. For more details, see Generating AI models.

Setting up training for log anomaly detection - natural language

Set up training for this AI algorithm to detect unusual patterns in your logs and notify you by generating an alert when they occur.

About this task

For more information on the how the different types of log anomaly detection work, see About log anomaly detection algorithms.

To set up training for this algorithm, you need log data that shows normal operational behavior. If your logs include anomalous situations, such as a critical incident, technical glitch, or some other change that deviates from the norm, you can filter that data out before training the algorithm.

Training setup of the algorithm is done by completing some setup tasks. When the training setup is complete, you need to go to a different part of the AI Model Management to train the model.

Note If both natural language and statistical baseline log anomaly detection algorithms are active, then any log anomalies that are discovered by both will be reconciled, so that only one alert is generated.

Before you begin

When providing log data, select a date range that includes at least 10,000 lines of messages for each resource that you want insights from. An ideal amount would be around 100,000 lines. When selecting longer date ranges, consider that it might take more time to process the data and might have system impacts.

To check that you have met these requirements, review the prerequisites for this AI algorithm. For more details, see the rows that are associated with Log anomaly detection in the prerequisites table on this page: Prerequisites.

In addition, you must perform the following tasks:

- Configure resources such as CPU and memory to ensure good training performance. For more details, see Configuring resources for log anomaly detection
- If you want to train on more than 14 days of log data, then you must increase the number of Elasticsearch shards. For more details, see Increasing the number of Elasticsearch shards.

Running the task

There are multiple parts to this task:

- Starting the training setup
- Checking data connections
- Specifying which data to train on
- Excluding irregular data
- Deciding how to run the training
- Deciding how to deploy the training
- What to do next
- Viewing the running jobs

Starting the training setup

Go to the AI Model Management and complete some setup tasks.

1. In the Cloud Pak for AIOps home page, click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **AI model management** to open the AI Model Management.
3. In the AI algorithms tab, click **Set up Training** within the **Log anomaly detection - natural language** tile.

Note: If this AI algorithm has already been set up for training, then the word Set up appears in grayed out text at the bottom of the tile. In this case, you can edit the algorithm training setup, as described in Editing the training setup for an algorithm.

Checking data connections

Review the information on this page. It explains what this AI algorithm is, how it helps in your production environment, and provides a list of data and tool connections that are needed to generate a model.

1. On the Data and tool connections, check that at least one connection is listed.

   Note: If no connections are listed, or you expected to see more connections, click **Go to Data and tool connections** to modify your connections. For more details, see this page: Prerequisites.

2. Click **Next** to move to the next panel.

Specifying which data to train on

Review and modify your data connections, if necessary.

1. Review the Data assets table.

   This table lists the data connections from the **Getting started** panel, and provides the following information for each data connection:

   - How is the data being collected? Options are Live or Historical.
   - Is the data flow on?

To make any changes, click **Go to Data and tool connections** to modify your connections. For more details, see this page: Prerequisites.

Optionally remove a data connection by clicking the checkbox to the left of the data connection name, and clicking the Delete icon that appears.

2. Enable or prevent incremental training.

BETA *Drifted resource detection* is available only for technology preview, and is not for production usage.
Incremental training prevents redundant training of preexisting models in consecutive training runs. With incremental training, each time that a training run launches, models are trained only on newly discovered resources and drifted resources, such as resources with changed log patterns. If neither of these resources are discovered in the training data since the previous training run, no new models are trained. Incremental training can significantly decrease training times.

When incremental training is prevented, models are retrained in each run to keep them up to date with the training data, but the training time increases.

3. Specify a date range to train on. Select **Preset** for a preset option or **Custom** for a range of dates.

Tip If you are planning to schedule the execution of the training for this algorithm then you must select one of the preset options that retrieves a fixed window of data relative to the current time.

- **Preset**: Click this option and select one of the following time ranges. Note that 14 days of log data is the maximum amount that is supported with the default values.
    - Past day
    - Past 3 days
    - Past 7 days
    - Past 10 days
    - Past 14 days

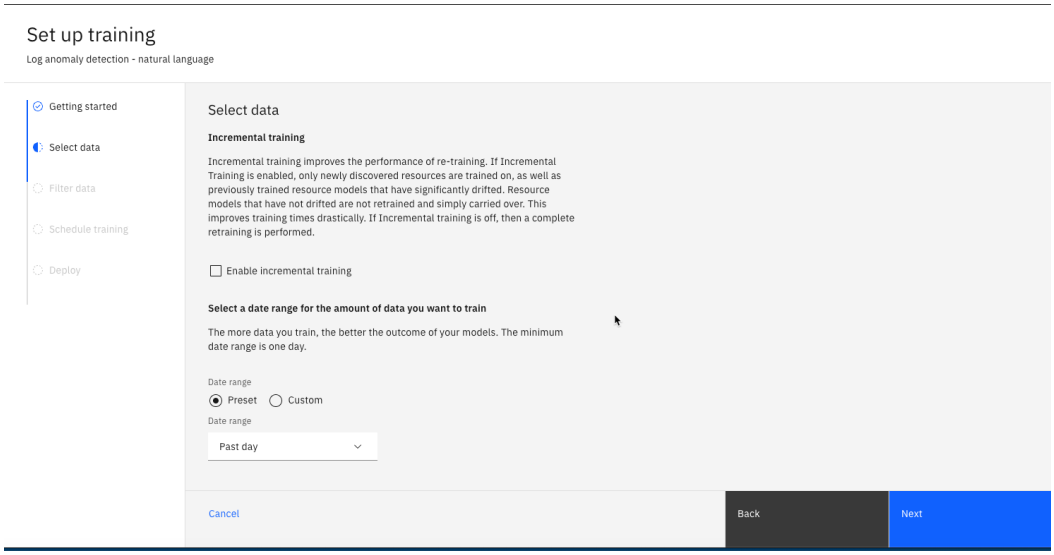


Figure. Schedule training

- **Custom**: This is an easy and quick option to use if you are not concerned with training models on data from a very specific window of time. Click this option and specify a fixed date range. Note that if you want to train on more than 14 days of log data, then you must increase the number of Elasticsearch shards. For more details, see Increasing the number of Elasticsearch shards.

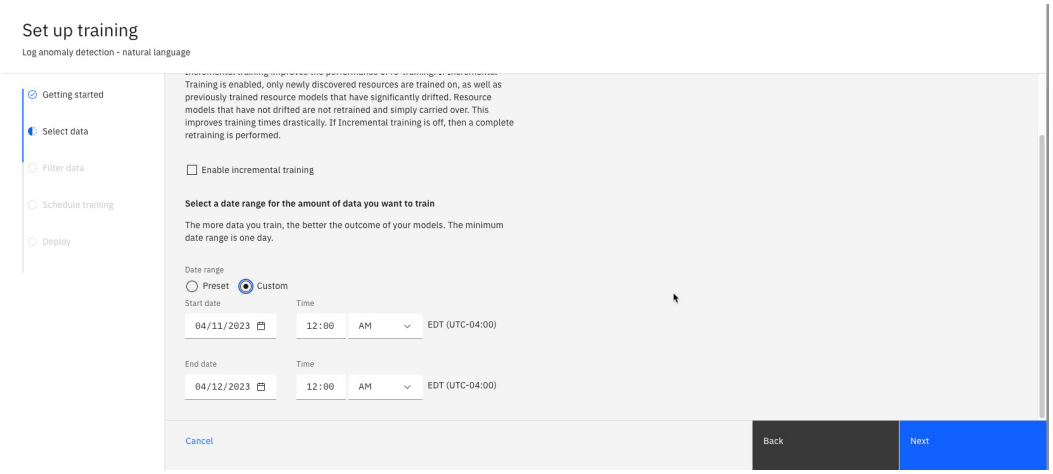


Figure. Custom

Note When you set custom dates and times, you must be aware of the following:

- You have the option to specify start and end times in local time or UTC. By default, the time is in local time.
- Ingested log data is stored in Elasticsearch in UTC.
- To ensure that you select the correct data intervals, specify both the start and end times in this panel in UTC.

For example, if you select November 15 12:00 AM through December 15 11:59 PM UTC time, the training window includes data from the start day of November 15 through to the end of the day on December 15. If you instead select December 15 12:00 AM, then the training window includes data from November 15 through December 14 11:59 PM.

- If you choose to use local time, you need to adjust the start and end date and time to match the period for which log data is available in ES.

4. Click **Next** to move to the next panel.

Filter data

Filter data from the training. This includes data associated with event storms, maintenance windows, major events, and other unusual outages. This will enable the Log anomaly detection algorithm to baseline normal behavior and record expected log messages.

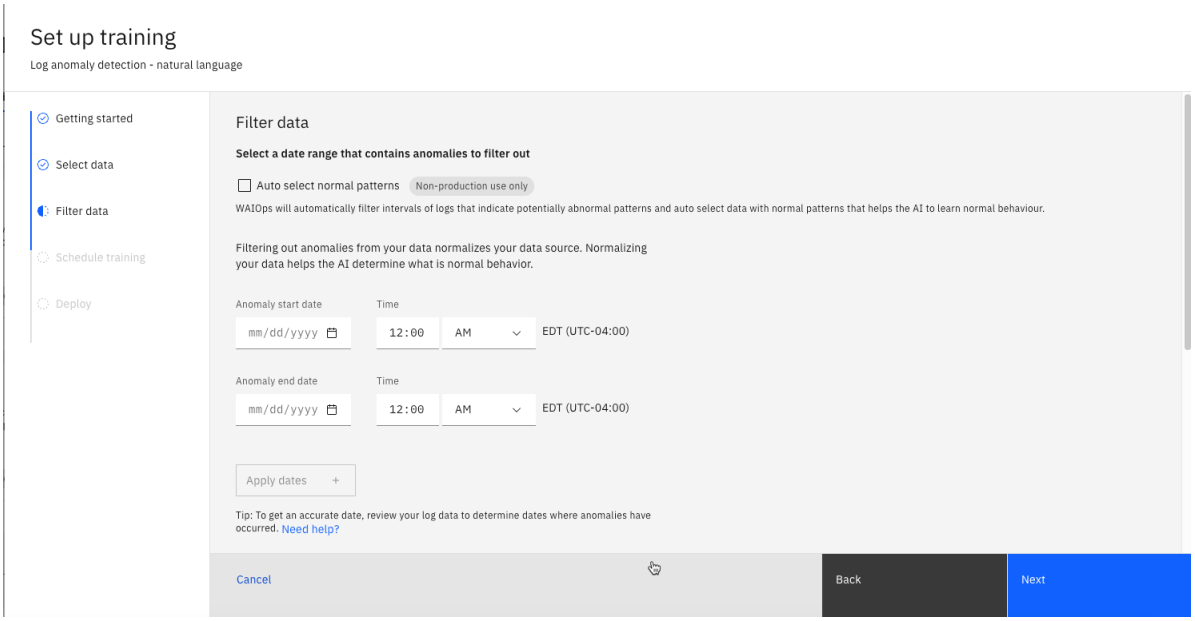


Figure. Log anomaly

1. (Optional) Select date range of irregular data to filter out. Then click **Apply dates**. Your selection appears in the **Applied dates** table below.

Tip Go to your data source to view your log data. This will help you determine when there were incidences of irregular data.

Note When you set custom dates and times, you must be aware of the following:

- You have the option to specify start and end times in local time or UTC. By default, the time is in local time.
- Ingested log data is stored in Elasticsearch in UTC.
- To ensure that you select the correct data intervals, specify both the start and end times in this panel in UTC.

  For example, if you select November 15 12:00 AM through December 15 11:59 PM UTC time, the training window includes data from the start day of November 15 through to the end of the day on December 15. If you instead select December 15 12:00 AM, then the training window includes data from November 15 through December 14 11:59 PM.
- If you choose to use local time, you need to adjust the start and end date and time to match the period for which log data is available in ES.

2. (Optional) Repeat step 1 for each date range of irregular data you want to apply.

   The data set is listed in the **Data sets** table at the bottom of the screen.

3. (Optional) Enable **Auto select anomalies** to filter anomalies from data and normalize the data source.

   BETA *Auto select anomalies* is available only for technology preview, and is not for production usage.

   Filtering anomalies helps the log anomaly detector distinguish between anomalous and normal behavior.

   For enhanced filtering, specify an **Anomaly start date** and an **Anomaly end date** to filter out anomalies that occurred during that time.

4. Click **Next** to move to the next panel.

Schedule the training

Decide whether to run the training setup on demand or to schedule it to run on an ongoing basis.

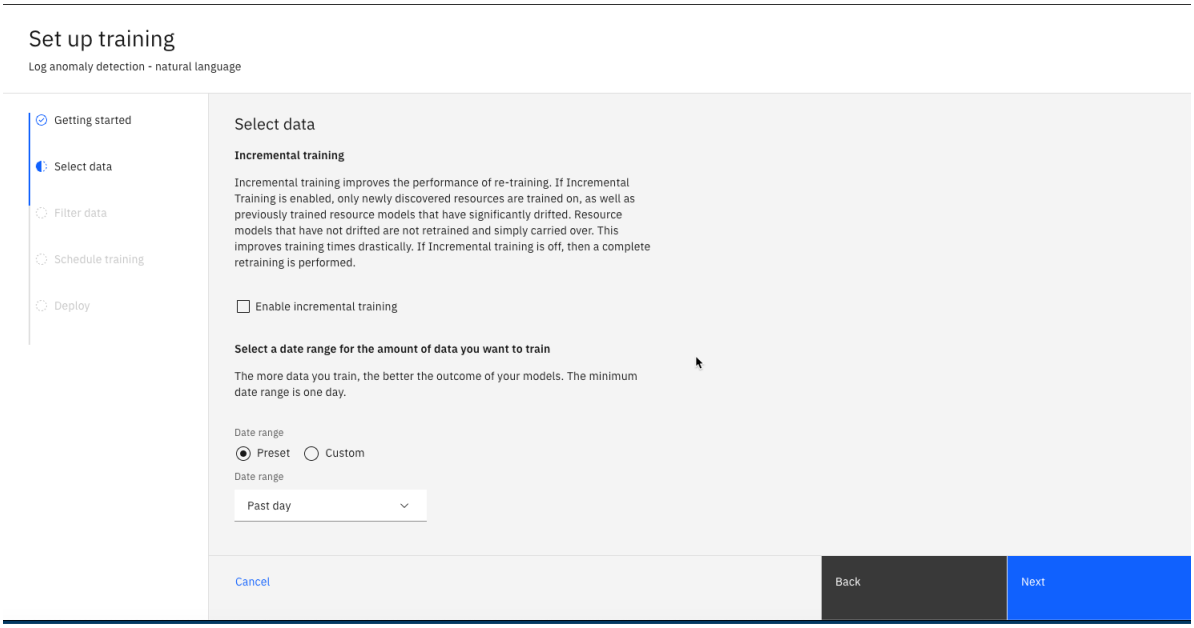


Figure. Schedule training

1. Proceed as follows:

   - To run the training setup on demand, ensure that **Schedule to run** is set to **Off**, and go to step 3.
   - To run the training setup on a schedule, set **Schedule to run** to **Yes**, and go to the next step.

2. Schedule the run. Click this option to specify a schedule. You can specify a start date with an optional end date, a frequency, and a time based on Coordinated Universal Time (UTC).

   Note:

   - 'Start' is when you want the scheduler to begin launching the training run. You can select 'now' for the scheduler to immediately start running, or you can specify a custom time by using the 'on date' option.
   - 'End' lets you specify how long you want the scheduler to manage the training runs. By selecting 'Never', the scheduler will keep running trainings (on the defined frequency) forever. You can also specify a custom end date by selecting 'on date'.
   - 'Frequency' lets you select how often they want the model to be retrained. The possible options are daily, weekly, bi-weekly, or monthly.
3. Click **Next** to move to the next panel.

Deciding how to deploy the training

Decide whether to review the results of training before deploying the model, or to have the system automatically deploy the model.

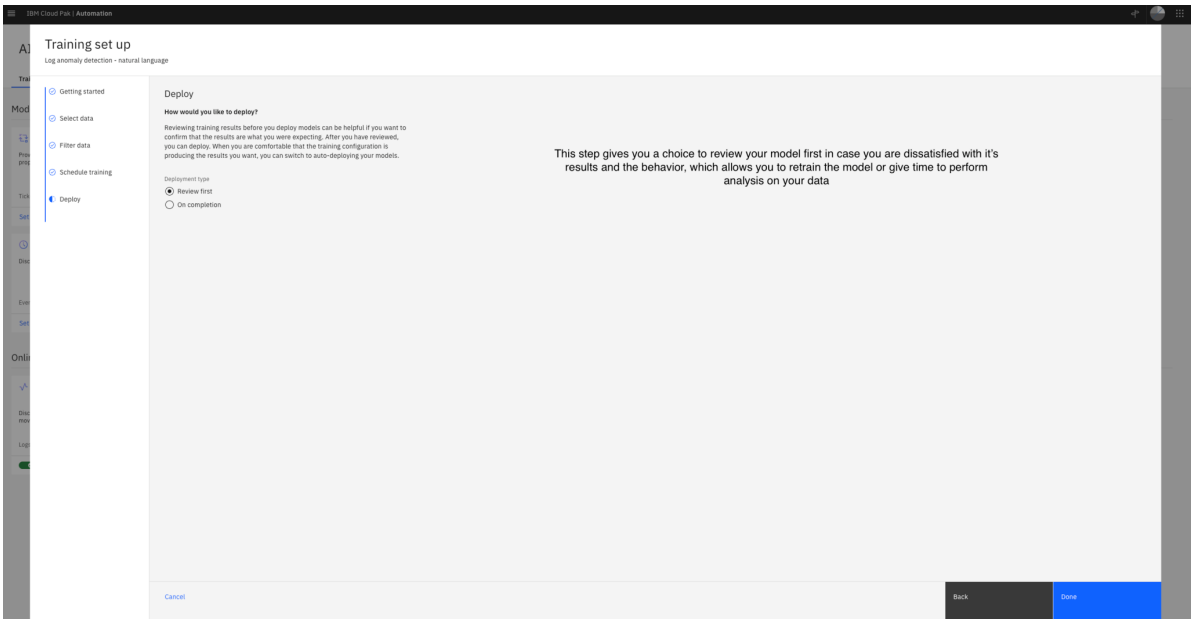


Figure. Deploy training

1. Proceed as follows:

   - To review the results of training before deploying the model, ensure that **Deployment type** is set to **Review first**.
   - To have the system automatically deploy the model, set **Deployment type** to **On completion**.
2. Click **Done** to save the training setup for the algorithm.

Viewing the running jobs

1. From the main navigation of the console, click **AI model management**.
2. Click on the tile of the AI algorithm you want to view.
3. From the **Summary** tab, click **Start training** to begin training your AI algorithm. While the training runs, the jobs and their statuses are displayed in the **Activity** section.

What to do next

Now that the training setup is complete, you can train the algorithm. For more details, see Launching the training.

Setting up training for similar tickets

Set up training for this AI algorithm to enable the discovery of historical tickets to aid in the remediation of current tickets.

About this task

When an incident occurs, it can be helpful to review details for similar tickets from the past to help determine a resolution. This model aggregates information from past similar tickets. It can also extract the steps that are used to fix previous tickets, if documented.

Training setup of the algorithm is done by completing some setup tasks. When the training setup is complete, you need to go to a different part of the AI Model Management to train the model.

Before you begin

To train similar tickets, you must provide ticket information. There is no set minimum of how much data to provide; the AI trains on whatever amount is available.

However, the more ticket data you can provide, the better. Any data that is collected for historical training is available to use as data. You can also collect data from the live connection and as future tickets are created, they are processed automatically.

Qualifying tickets

Important: Starting with IBM Cloud Pak® for AIOps version 4.1.0, similar incidents, such as from a ServiceNow integration, are called similar tickets.

To pass data precheck, a minimum of five qualifying tickets are needed. Qualified tickets are closed:

- The `state` and `closed_at` fields from the raw ticket schema indicate closure.
- The `close_notes` field from the raw ticket schema includes meaningful resolution notes with actions and entities that the similar tickets algorithm can extract. If your tickets do not include resolutions, or if they include default entries like *Closed by Caller*, these values do not identify paths to remediation, and the similar tickets algorithm is unable to use the tickets to provide recommended actions through similar tickets.

The following required fields are needed for a ServiceNow ticket to be qualified for the similar tickets AI algorithm.

Required attributes for the similar tickets algorithm

Attribute	Description
`short_description`	Short description or title of the ticket.
`number`	Identification number of the ticket.
`opened_at`	Coordinated Universal Time at which the ticket started formatted as `yyyy-mm-ddTHH:MM:SSZ`.
`state`	Current state of the ticket.
`closed_at`	Coordinated Universal Time at which the ticket was closed formatted as `yyyy-mm-ddTHH:MM:SSZ`.
`close_notes`	Comments or closed notes that talk about the list of actions that are taken, or comments that are added by the user who is attending to the ticket.

Running the task

There are multiple parts to this task:

- Starting the training setup
- Checking data connections
- Deciding how to run the training
- Deciding how to deploy the training
- What to do next

Starting the wizard

Go to the AI Model Management and complete some setup tasks.

1. In the Cloud Pak for AIOps home page, click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **AI model management** to open the AI Model Management.

3. On the **Training** tab, click **Set up Training** within the **Similar tickets** tile under the Trainable AI algorithms section.

Checking data connections

Review the information on this page. It explains what this AI algorithm is, how it will help in your production environment, and provides a list of data and tool connections needed to generate a model.

1. On the Data and tool connections, check that at least one connection is listed.

   Note: If no connections are listed, or you expected to see more connections, click **Go to Data and tool connections** to modify your connections. For more information, see ServiceNow connection.

2. Click **Next** to move to the next panel.

Schedule the training

Decide whether to run the training setup on demand or to schedule it to run on an ongoing basis.

1. Proceed as follows:
   - To run the training setup on demand, ensure that **Schedule to run** is set to **Off**, and go to step 3.
   - To run the training setup on a schedule, set **Schedule to run** to **Yes**, and go to the next step.
2. Schedule the run. Click this option to specify a schedule. You can specify a start date with an optional end date, a frequency, and a time based on Coordinated Universal Time (UTC).
   - Decide whether to run the training setup on demand or to schedule it to run on an ongoing basis.
   - Training loads existing qualified ServiceNow tickets into the similar tickets AI algorithm.
   - If you set the ServiceNow connection to continuous data collection, scheduling frequent trainings helps ensure that the similar tickets AI algorithm includes the new tickets in its recommendations.
3. Click **Next** to move to the next panel.

Deciding how to deploy the training

The option to review training results before deployment is unavailable for this algorithm. After training is complete, the model is automatically deployed. To see similar tickets appear in your subsequent incidents, go to the **Automations** page and enable **Default story query similar incidents service policy** in the policy list.

Click **Done** to save the training setup for the algorithm.

What to do next

Now that the training setup is complete, you can train the algorithm. For more details, see Launching the training.

Setting up training for temporal grouping

Set up temporal grouping to train policies. Then, choose how to deploy the policies.

For more information about the temporal grouping AI algorithm, see About temporal grouping.

Before you begin

To train a definition for temporal grouping, this AI algorithm needs at least three months of historical event data to learn alert patterns and discover groups. A set of data creates a set of policies. For more information, see Prerequisites.

This AI algorithm can start to discover groups after a pattern is learned, which might take hours or days. For example, if a problem occurs in the first and the fourth week of January and again in March, the algorithm learns to group the alerts that tend to co-occur when the problem happens.

About this task

There are multiple parts to this task:

- Starting the wizard
- Scheduling the training
- Deciding how to deploy the training
- Running the training again
- What to do next

Starting the wizard

Go to the AI Model Management and complete some setup tasks.

1. In the Cloud Pak for AIOps home page, click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **AI model management** to open the AI Model Management.
3. In the **Training** tab, click **Set up training** within the **Temporal grouping** tile under the Trainable AI algorithms section.

Scheduling the training

Decide whether to run the training set up on demand or to schedule it to run on an ongoing basis.

1. Proceed as follows:
   - To run the training set up on demand, ensure that **Schedule to run** is set to **Off**, and go to step 3.
   - To run the training set up on a schedule, set **Schedule to run** to **Yes**, and go to the next step.
2. Schedule the run. Click this option to specify a schedule. You can specify a start date with an optional end date, a frequency, and a time based on Coordinated Universal Time (UTC).
3. Click **Next** to move to the next panel.

Deciding how to deploy the training

The option to review training results before deployment is disabled for this algorithm. However, you can choose when to deploy the policies that are generated by training the algorithm.

- To review and manually deploy the model, set **Deployment type** to **Review and manually deploy policies**.
- To automatically deploy the model, set **Deployment type** to **Automatically deploy all policies**.

Click **Done** to save the training setup for the algorithm.

Running the training again

- If you run the training again with the same data, expect to receive the same policies.
- If you run the training again with different data, you might receive new policies.

What to do next

Now that the training setup is complete, you can train the algorithm. For more details, see Launching the training.

Managing AI algorithms

Use this information to understand how to generate AI models, and how to perform various administration tasks on the models.

For more information about managing AI algorithms, see the following topics:

- Generating AI models
- Deploying AI models
- Editing algorithm configurations
- Deleting algorithm configurations
- Managing AI model versions
- Draft and archived policies

Generating an AI model

Generate an AI model by training one of the AI algorithms. The AI model that is generated by the training acts on live data and generate insights to your SREs.

About this task

You only need to generate an AI model for trainable AI algorithms. Pre-trained AI algorithms, in contrast, analyze your data in real time and derive insights without the need to generate a model. For more details, see Algorithm types.

Before you begin

You must save at least one algorithm training setup, as described in Setting up training for trainable AI algorithms.

Procedure

1. In the Cloud Pak for AIOps home page, click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **AI model management** to open the Training page.

   A list of algorithms is shown in tile format under sections depending on whether they are trainable or pre-trained AI algorithms.
3. Click 'Set up training' on the trainable algorithm tile for which you want to generate an AI model. When the setup is complete for that algorithm, the details from the 'Training Set up' page appear. To understand a particular model - what it does and what kind of data you need to run it – click *Description* (information icon) next to the tile's title.

   Note: You can run precheck before training your model. Precheck checks the quality of the data first. Regardless, if you try to train models without first running the precheck, AI Models still run the precheck first anyhow. Even if the training data changes, the precheck will be rerun on the new data.
4. If the setup is complete for an algorithm, click the algorithm's tile for more details on that AI model and to begin the training.

   When the model detail page opens, you can click **Precheck data** from the panel on the right side of the screen, before training. This checks the quality of the data. Regardless, if you try to train models without running the precheck, AI models still run the precheck first, anyhow. Even if the training data changes, the precheck will be rerun on the new data. **Note**: There is no data precheck for Metric anomaly detection.
5. In the right side panel, click **Train Models**.

   After a few moments the IBM Cloud Pak Automation console responds as follows:

   - A message appears with the text: **Training successfully started**.
   - The **AI Training** tile displays the message: **In Progress**. Training is now in progress.
6. As training progresses it passes through phases, which you can view under *Activity*, in the side panel.

There are six possible phases that are shown, though they vary depending on which algorithm you chose to train:

```
1. **Queued**
2. **Data retrieving**
3. **Quality check**
4. **Preparing data**
5. **Training**
6. **Saving models**
```

Only change risk contains all six phases. Once training is complete, the **AI Training** tile displays the messages:

- **Complete**
- **Deployed** (if indeed it has also been deployed).

At this point AI models have been created, with a specified data quality, which can take one of the following values:

- **Good**
- **Needs improvements**

  Note: No quality check is carried out on the Metric anomaly detection or Temporal Grouping models, as the check is not supported by the API that provides data to these algorithms.

For more details, click the information symbol in the **Data quality** tile. For a full set of data quality measures and messages, see Data quality messages.

For log anomaly detection training, an extra **Models** tile is displayed containing information on the models that were generated during training. For more details, see Displaying log anomaly detection resources and models.

Change risk tiles

For **change risk**, there is a new section entitled **Ticket data quality**, consisting of three tiles:

- *Closed tickets*: A number. For example, 10000. Historical ticket data must be in a terminal state (closed tickets) in order for the algorithm to train.
- *Missing tickets*: Data in high-importance fields within a ticket is needed to create good models. The more complete the data that you have in all fields (such as type, business service, sys domain and short description), the better your models perform. **Note:** if less than 0.5% of the total closed tickets are missing the relevant field (type, business service, and so on), it is displayed as 0% in the UI.
- *Ticket Word Count*: Tickets with a short (<5 words) overall word count generally don't provide enough information for the algorithm to create good models. This field combines long and short count words for the number of tickets, say 10000.

There is also a **Model prediction** tile, which shows the number of incidents causing tickets as a percentage.

What to do next

If you selected manual deployment when setting up training for the model, then you can now proceed to deploy the AI models that you created. For more details, see Deploying models.

If you encountered errors in the training, see the following sections for more information:

- Troubleshooting documentation at Troubleshooting AI model management.
- Release notes at Known issues and limitations.

Reference information

The following information is referenced in this task.

List of algorithm training setups

Column	Description
AI algorithm	The AI algorithm that is related to this record. Possible values are: **Change risk**, **Log anomaly detection**,**Metric anomaly detection**, **Similar incidents**, and **Temporal grouping**.
Latest model version	Latest version of the model. A new version of the model is generated every that time the algorithm training setup is run.
Version deployed	This is the currently deployed version of the model. If training is scheduled and deployment is automatic, then the value in this field always matches the value in the **Version** field.
Date trained	Date and time this training setup was last run. If it has never been run then this field is blank.
Activity	Displays one of the following status values: (a) **Not started**: this training setup has not yet been run; (b) **Training**: the training setup is currently running; (c) **Training queued**: training is currently waiting on another training job to complete; (d) **Training complete**: the most recent training setup run was successful; (e) **Failed**: Training failed.

Column	Description
Danger zone Contains the **Delete** option, which will delete all models for the algorithm.	

Displaying log anomaly detection resources and models

During natural language log anomaly detection, models are produced for each of the resources identified in the log data provided as input to the algorithm. On completion of log anomaly detection training, the system provides information on the models that were generated during training.

About this task

In order to produce a model for a resource, a minimum of 2,000 log lines is required for that resource and the log data must meet formatting requirements. On completion of training, a **Models** tile is displayed containing the details described below.

Models

The main part of the tile displays a donut graph showing resources with models in green and resources without models in red.

- Total number of models identified is shown in the middle of the donut graph.
- Hover over the donut graph to display the number of resources with models and the number of resources without models.

View resources

Click this link at the top right of the **Models** tile to display resource and model details.

Column	Description
Resource name	Name of the resource for which models were produced.
Log lines found	Number of log lines found for this resource in the data supplied when configuring the algorithm. A minimum of 2,000 log lines is required per resource to meet data sufficiency requirements. **Note** The number of log lines displayed here might be smaller than the number of log lines provided by the log data connection and presented in a log aggegator dashboard. This is because log lines are deduplicated during the training process. A log line is said to be duplicated when the combination of **timestamp**, **instance_id** and **message** is exactly the same. The duplicate log lines are not saved on Elasticsearch and are also not used for inference in log anomaly detection. Deduplication has no impact on the quality of the models generated but does have the advantage of lower data storage requirements and also helps to manage Flink restart issues due to collection of the same data multiple times.
Sufficient data	Displays one of the following values: **Passed**: 2,000 log lines or more were found for this resource. **Failed**: Less than 2,000 log lines were found for this resource.
Data format	An indication of whether the data found for this resource meets data formatting needs. This column takes one of the following values: **Passed**: Formatting needs are met. **Failed**: Formatting needs not met. You might need to modify the field mapping of the data. For more information see Mapping data.
Model status	Indicates whether a model was created for this resource. A model is only created if both sufficiency and formatting needs are met.

Data quality messages and alerts

The data quality precheck stage always runs on your input data before it can be approved for training. Once the precheck stage is complete, you are presented with an assessment of the data quality.

The data quality precheck examines whether enough data is available to return meaningful results (*data sufficiency*), and whether the data meets mapping requirements (*data formatting*). A result is displayed on the console, and this result affects the execution of the model training, as if the precheck fails, the training is disabled, and you cannot run training.

If you change the selected training data, the precheck stage is always run again.

Mapping data

Poor *data formatting* is often associated with incorrect field mapping.

Data is collected when you select a source on the Data and tool connections and add a connection. Within the window for configuring the connection, use the **Field mapping** section to map source fields to standard fields. For example, the `_timestamp` in LogDNA data should be mapped to `timestamp` in the standard format. One reason that schema validation might fail, is when incorrect data mappings are entered in the **Mapping** field. For more information on the overall data loading and AI training process, see Planning data loading and training.

Good data quality

Good data quality means that requirements for *data sufficiency* and *data formatting* are met. Good data quality is represented on the Automation console using the following label: **Good data**.

Data quality	Recommendation	Description
Good data	`Precheck came back with good data. You are now ready to start training models.`	Requirements for *data sufficiency* and *data formatting* are met.

Poor data quality

Poor data quality messages vary with the AI algorithm that you are training. Poor data quality is represented on the Automation console using the following label: **Data needs improvement**.

For each of the following AI algorithms, this topic describes the possible data quality messages, provides a brief description of the message, and points to any relevant documentation task, which can support remediation action.

- Change risk
- Log anomaly detection
- Similar tickets

Change risk

This algorithm takes as input change and incident data from ServiceNow. To meet overall data quality requirements, both change and incident data must meet *data sufficiency* and *data formatting* requirements.

Data quality alert	Recommendation	Description	Link
Data needs improvement	`Part of your data did not pass precheck. View recommendations to see how you might improve data quality.`	**Data precheck issue**: Insufficient number of problematic change tickets for a good model. For a closed change ticket count less or equal to 150, you need at least three problematic tickets. For a closed change ticket greater than 150, you need at least two percent of change tickets to be problematic. A problematic change ticket is defined as a change ticket that can either be associated with incident tickets, or have an 'unsuccessful' value set in the change ticket close code field.	See ServiceNow connection
Data needs improvement	`Change data: The data is not valid according to schema. Check input data mapping or data source.`	**Data formatting issue**: Change data does not meet *data formatting* needs. You might need to modify the data's field mapping.	See ServiceNow connection

Data quality alert	Recommendation	Description	Link
Data needs improvement	`More data is needed. You need at least _X_ change tickets for a good model.`	**Data sufficiency issue**: Change data does not meet *data sufficiency* needs. More change ticket data is needed	See ServiceNow connection
Data needs improvement	`Incident data: The data is not valid according to schema. Check input data mapping or data source.`	**Data formatting issue**: Incident data does not meet *data formatting* needs. You might need to modify the data's field mapping.	See ServiceNow connection
Data needs improvement	`More data is needed. You need at least 1 incident data point for a good model.`	**Data sufficiency issue**: Incident data does not meet *data sufficiency* needs. More change ticket data is needed	See ServiceNow connection
Data needs improvement	`Data is insufficient for a good model. You need at least 2 percent of change tickets to either be associated with incident tickets or have an 'unsuccessful' value set in the change ticket close code field.`	**Data sufficiency issue**: Incident data does not meet *data sufficiency* needs. You must provide data with a change ticket failure rate of 2% or higher.	See ServiceNow connection

Log anomaly detection

This algorithm takes as input log data from one or more log systems. The log anomaly detection algorithm itself has multiple underlying microservices. *Data sufficiency* and *data formatting* requirements for this algorithm work as follows:

- Each of these microservices requires a minimum of 2000 data points. To meet the minimum *data sufficiency* requirement for the algorithm as a whole, at least 75% of the algorithms must meet this 2000 data point requirement. For example, if only two microservices out of 10 have less than 2000 data points, and the rest have more than 2000 data points, then *data sufficiency* requirement for the algorithm as a whole is good.
- To meet the *data formatting* requirement for the algorithm as a whole, at least 75% of all data (from all of the microservices) must be valid with respect to the schema.

Data quality alert	Recommendation	Description	Link
Data needs improvement	`Part of your data did not pass precheck. There were not enough logs for X out of Y resources. Models were only created for those where sufficient data existed. You can expand the date range to gather more data and train again to expand coverage.`	**Data precheck issue**: Data does not meet data precheck needs. You might need to modify the data's quality before running the training.	See Train model precheck
Data needs improvement	`There were not enough logs for _X_ out of log_anomaly_detection components. Models were only created for those where sufficient data existed. You can expand the date range to gather more data and train again to expand coverage.`	**Data sufficiency issue**: Data does not meet *data sufficiency* needs. More than 25% of the component microservices have less than the required 2000 data points. More log data is needed. Try expanding the data range of the data provided.	See Natural language log anomaly detection: data to train on
Data needs improvement	`We have detected that a portion of this data set is in an unsupported language, and could affect the quality of this model. We recommend that you remove this data from the data set and retrain the model.`	**Language support issue**: Log anomaly detection only supports the following languages and language combinations: English, Spanish, German, English and Spanish, English and German. Your data contains either an unsupported language or an unsupported language combination.	For more information, see Language support. To remediate this issue, see Managing unsupported languages.

Similar tickets

This algorithm takes as input incident data from ServiceNow. To meet overall data quality requirements, both change and incident data must meet *data sufficiency* and *data formatting* requirements.

- To meet *data sufficiency* requirements, a minimum of five closed incidents with associated resolution is required.

Data quality alert	Recommendation	Description	Link
Data needs improvement	`Part of your data did not pass precheck. View recommendations to see how you might improve data quality.`	**Data precheck issue**: More data is needed. You need at least five closed incidents with resolution for a good model.	See ServiceNow connection
Data needs improvement	`More data is needed. You need at least 5 closed incidents with resolution for a good model.`	**Data sufficiency issue**: Data does not meet *data sufficiency* needs. More closed incident ticket data is needed.	See ServiceNow connection

Note There is a new data quality alert that appears in the console if the precheck is launched (for all of the three algorithm failures) without any data loaded. The alert tells you that the 'Precheck failed', and the recommendation is 'Cannot find any data'.

Managing unsupported languages

If natural language log anomaly training reports that the model needs improvement because of unsupported language issues, you can remove the offending data from the data set by deleting component-specific logs, and then retrain.

About this task

Log anomaly detection only supports the following languages and language combinations:

- English
- English and French
- English and German
- English and Italian
- English and Spanish
- French
- German
- Italian
- Spanish

If your data contains either an unsupported language or an unsupported language combination, then you will get the following data quality alert on completion of training:

Needs improvement We have detected that a portion of this data set is in an unsupported language, and could affect the quality of this model. We recommend that you remove this data from the data set and retrain the model.

To remediate the situation you must delete all the logs from components whose logs are all or partially in an unsupported language.

Procedure

Perform the following steps to remediate the situation.

1. Log in to your cluster.

   ```
   oc login -u kubeadmin -p <password>
   ```

   For more information, see Logging in to the OpenShift CLI.

2. Run the `oc project` command to set the context to the project where Cloud Pak for AIOps is deployed.

3. Find the name of the apiserver pod.

```
oc get pod | grep api-server
```

4. Open a terminal on that pod.

```
oc exec -it <api-server_podname> bash
```

5. Run the following **curl** command exactly as specified below to retrieve data quality alert details.

```
curl -k -u $ES_USERNAME:$ES_PASSWORD -X GET
"https://$ES_ENDPOINT/prechecktrainingdetails/_search?pretty" -H 'Content-Type:
application/json' -d'{"_source":"dataQualityDetails.languageInfo.components"}'
```

In this command:

- **ES_USERNAME** is the Elasticsearch username.
- **ES_PASSWORD** is the Elasticsearch password.
- **$ES_ENDPOINT** is the Elasticsearch environment variable in the pod.
- **https://$ES_ENDPOINT/prechecktrainingdetails** is the Elasticsearch endpoint for the **prechecktrainingdetails** index, which stores language information associated with the data set sampled for a specific component.

6. In the output that is generated, navigate to the **dataQualityDetails** field, and find the nested **languageInfo** field. Then retrieve the names of the component where the language list has **unknown** as one of the values. For example, in the example below you would select **abc**.

```
"components" : [
              {
                "name" : "abc",
                "language" : [
                    "English",
                    "unknown"
                              ],
```

7. For each component that you identified in the previous step, delete the logs from all log data indices by running the following command.

```
curl -k -u $ES_USERNAME:$ES_PASSWORD -X POST "https://$ES_ENDPOINT/*-
logtrain/_delete_by_query" -H 'Content-Type: application/json' -d'{"query": {"bool":
{"must": [{"term": {"instance_id.keyword": "abc"}}]}}}'
```

Where **abc** must be replaced with the name of the component.

Note If all components are detected as having logs in an unsupported language, then this step will delete all data from the Elasticsearch indices for log anomaly training. You will need to train again when fresh data arrives.

8. Once you have deleted the relevant logs, you must relaunch the training of your natural language log anomaly detection algorithm, as described in Generating AI models.

Determining which algorithm generated a log anomaly

There are two log anomaly detection AI algorithms, each of which can run independently of the other: natural language and statistical baseline. If both algorithms are enabled, then any log anomalies discovered by both will be reconciled, so that only one alert is generated. Use the information in this topic to work out which log anomaly detection algorithms contributed to the detection of the anomaly.

About this task

There are two ways to check which log anomaly detection AI algorithms generated a given log anomaly:

- By viewing the details of the alert in the Alert Viewer.
- By monitoring the Kafka topic **cp4waiops-cartridge.lifecycle.input.events kafkatopics**, and parsing out the values in the **log_anomaly_model** field.

Viewing details in the Alert Viewer

1. In the Alert Viewer, double click the log anomaly alert of interest. The **Alert Details** panel opens on the right, displaying the properties of the alert.
2. In the **Properties** table, locate the **Description** field.
3. Within the **Description** field, locate the **log_anomaly_model** key. The nested values within this key identify which log anomaly detection algorithm contributed to the detection of this anomaly. Possible nested values are listed in Possible values within the log_anomaly_model JSON key below.

Monitoring the relevant Kafka topic

1. Monitor the Kafka topic **cp4waiops-cartridge.lifecycle.input.events kafkatopics**. This topic provides a Kafka stream of alert instances.
2. Look for a **sender** key where name attribute is "Log Anomaly" to identify those alert instances that were generated by the log anomaly detection AI algorithms.
3. Within that section of the JSON, locate the **log_anomaly_model** key. The nested values within this key identify which log anomaly detection algorithm contributed to the detection of this anomaly. Possible nested values are listed in Possible values within the log_anomaly_model JSON key below.

Reference information

The following information is referenced in this task.

Possible values within the log_anomaly_model JSON key

Value	Algorithm	Meaning
PCA_Stats	Natural language	The time window in which this alert occurred is statstically different to the training data.
PCA_error_or_unknown_template	Natural language	There are error templates or unknown templates in this time window which the algorithm has identified as anomalous.
PCA_low_frequency_error_template	Natural language	There are error templates which occur with low frequency in this time window and the algorithm has identified this as anomalous.
RSM_entities	Statistical baseline	The algorithm has determined that there are significantly different error entities in this time window.
RSM_embeddings	Statistical baseline	The algorithm has found word embeddings for this time window that are distant from the embedding vectors for normal data.
RSM_websphere	Statistical baseline	This value is specific to WebSphere logs. The algorithm has found significantly different error message identifiers or log levels for this time window.

Deploying models

If the model is not autodeployed, then if you are happy with the results, you can now deploy the model.

Before you begin

You only need to perform this task if you generated a model that requires manual deployment.

Procedure

1. From the AI Model Management, click the options menu for the algorithm configuration that includes the model that you want to deploy.
2. Click **Deploy**.

Reference information

The following information is referenced in this task.

List of algorithm configurations

Column	Description
Version	Latest version of the model. A new version of the model is generated every time the algorithm configuration is run.
Deployed version	Currently deployed version of the model. If training is scheduled and deployment is automatic, then the value in this field always matches the value in the **Version** field.
AI algorithm	The AI algorithm related to this record. Possible values are: **Change risk**, **Log anomaly detection**, **Similar tickets**, and **Temporal grouping**.
Schedule	Displays the schedule frequency (**Daily**, **Weekly**, **Biweekly**, or **Monthly**). If no schedule was set, then this field displays the text **Run manually**.
Last trained	Date and time this configuration was last run. If it has never been run then this field is blank.
Status	Displays one of the following status values: (a) **Not started**: this configuration has not yet been run; (b) **Training**: the configuration is currently running; (c) **Training queued**: training is currently waiting on another training job to complete; (d) **Training complete**: the most recent configuration ran was successful; (e) **Failed**: Training failed.

Editing the training setup for an algorithm

You can edit the training setup for any of the existing algorithms.

Before you begin

You must save at least one algorithm training setup, as described in Setting up training for trainable AI algorithms.

About this task

Only algorithms that generate models can be edited. For more details, see Algorithm types.

Procedure

1. From the AI Model Management, under the Training tab, click the tile of the algorithm you want to edit.

   - **Change risk**: Deployment, Training schedule
   - **Log anomaly detection - natural language**: Training data, Deployment, Training schedule
   - **Similar tickets**: Training schedule
   - **Temporal grouping**: Training schedule
   - **Metric Anomaly Detection**: Training schedule

2. Modify the algorithm settings by clicking the **Edit data** icon  next to the settings that you want to modify. After you make your changes click **Save** to return to the Overview page.

   Your changes are now saved.

Reference information

The following information is referenced in this task.

List of algorithm details

Column	Description

Column	Description
Latest model version	Latest version of the model. A new version of the model is generated everytime that the training succeeds.
Version deployed	Currently the deployed version of the model. If training is scheduled and deployment is automatic, then the value in this field always matches the value in the **Version** field.
AI algorithm	The AI algorithm that is related to this record. Possible values are: **Change risk**, **Log anomaly detection - natural language vs statistical baseline**, **Similar tickets**, **Metric anomaly** and **Temporal grouping**.
Date prechecked	Date and time this training setup was prechecked.
Date trained	Date and time that this training setup was last run. If it has never run then this field is blank.

Deleting the training setup for an algorithm

You can delete the training setup for any of the existing algorithms.

About this tasks

If an algorithm training setup has associated AI models, then the action of deleting the training setup for the algorithm automatically removes any currently deployed model generated based on that training setup. It then deletes all of the models associated with this training setup.

Before you begin

You must have saved at least one algorithm training setup, as described in Setting up training for trainable AI algorithms.

Procedure

1. From the AI Model Management, click the options menu for the algorithm training setup that you want to delete.
2. Click **Delete**.

Reference information

The following information is referenced in this task.

List of training setups for an algorithm

Column	Description
Version	Latest version of the model. A new version of the model is generated every time the training setup for the algorithm is run.
Deployed version	Currently deployed version of the model. If training is scheduled and deployment is automatic, then the value in this field always matches the value in the **Version** field.
AI algorithm	The AI algorithm related to this record. Possible values are: **Change risk**, **Log anomaly detection**, **Similar tickets**, and **Temporal grouping**.
Schedule	Displays the schedule frequency (**Daily**, **Weekly**, **Biweekly**, or **Monthly**). If no schedule was set, then this field displays the text **Run manually**.
Last trained	Date and time this training setup was last run. If it has never been run then this field is blank.
Status	Displays one of the following status values: (a) **Not started**: this training setup has not yet been run; (b) **Training**: the training setup is currently running; (c) **Training queued**: training is currently waiting on another training job to complete; (d) **Training complete**: the most recent training setup ran was successful; (e) **Failed**: Training failed.

Managing AI model versions

You can change the version of a model that is deployed. You can also delete a model version.

For more information about managing AI model versions, see the following topics:

- Changing the deployed version
- Deleting model versions

Changing the deployed version of a model

Only one version of a model can be deployed at any one time. Where you have multiple versions of a model, you can change the deployed version.

Before you begin

You must have generated at least two versions of an AI model, and deployed one of them as described in Deploying models.

Procedure

1. In the AI Model Management, under *Trainable AI algorithms*, click the tile of the configuration you want to change.
2. When the overview page opens, click the *Model versions* tab. A list of model versions is displayed. For more information, see the List of model versions.
3. To change the deployed version, click the options menu at the far end of the table in the row corresponding to the version that you want to deploy. Then, click **Deploy**. The table updates to show that the version that you selected is now the deployed version.

Reference information

The following information is referenced in this task.

List of algorithms configurations

Column	Description
Name	Name of the algorithm configuration.
Latest model version	Latest version of the model. A new version of the model is generated every that time the algorithm configuration is run.
Version deployed	This is the currently deployed version of the model. If training is scheduled and deployment is automatic, then the value in this field always matches the value in the **Version** field.
AI algorithm	The AI algorithm that is related to this record. Possible values are: **Change risk**, **Log anomaly detection**, **Similar tickets**, and **Temporal grouping**.
Data prechecked	Date and time this configuration was last prechecked.
Date trained	Date and time this configuration was last run. If it has never been run, then this field is blank.

List of model versions

Note: The below columns are common to most algorithms and then they can vary in terms of additional columns. For example, Log anomaly also has 'Totals resources found', and 'Models created', whereas Change risk also includes 'Closed tickets' and 'Incidence avoidance'.

Column	Description

Column	Description
Model	Version of the model. A new version of the model is generated every that time the algorithm configuration is run.
Status	The deployed version is marked as **Deployed** with a comforting green tick to the left of the word **Deployed**. The other versions have the status **Complete**.
Last trained	Date and time when this version of the model was trained.
Duration	Length of time taken to train this version of the model.
Data Quality	Possible values are **Good** and **Needs improvement**. Only deploy model versions that have the status **Good**.

Deleting model versions

You can delete versions of a model.

About this task

You cannot delete a deployed version of the model directly. If you want to delete a deployed version, you must first change the deployed version of the model, and then delete the original model.

Procedure

1. From the AI Model Management **Training** tab, click **Log anomaly detection - natural language**.
2. From the **Log anomaly detection - natural language** page, click the **Model versions** tab.
3. To delete a model version, go to the row of the version that you want to delete and click the options menu to expand it. Then, click **Delete**.

   Note: You cannot delete a deployed version of the model directly. If you want to delete a deployed version, you must first change the deployed version of the model and then delete the original model.

Resetting the log anomaly training environment

If you need to reset your log anomaly training environment, complete the following steps. Proceed with caution and do not attempt these steps on a production or critical Cloud Pak for AIOps environment.

1. Run the `oc project` command to set the context to the project where Cloud Pak for AIOps is deployed.
2. Find the name of the `api-server` pod.

   ```
   oc get pod | grep api-server
   ```

3. Open a terminal on that pod. Replace the `<api-server_podname>` variable in the following command with the name of the `api-server` pod.

   ```
   oc exec -it <api-server_podname> bash
   ```

4. Open another terminal and run port forwarding for Elasticsearch.

   ```
   while true; do oc port-forward $(oc get po | grep iaf-system-elasticsearch-es-aiops-0 |
   awk '{print $1}') 9200; done
   ```

5. Get the Elasticsearch credentials.

   ```
   export username=$(oc get secret iaf-system-elasticsearch-es-default-user -o jsonpath="
   {.data.username}"|base64 -d);
   export password=$(oc get secret iaf-system-elasticsearch-es-default-user -o jsonpath="
   {.data.password}"|base64 -d);
   ```

6. **Important:** Do not delete any of the following index files. Deleting these files might delete data that corresponds to other AI activities, such as metric anomaly detection training. However, you can use `_delete_by_query` to delete documents that are referenced by these index files.

    - `trainingdefinition`
    - `trainingsrunning`
    - `trainingrun`
    - `trainingstatus`

7. Specify the variables and delete the following index files. Deleting an index uses a cascading delete and deletes the documents that are referenced by the index.

    - Any `1000-1000-*-logtrain` index files
    - The `1000-1000-vN-*` or `1000-1000-v*-models` index files of the particular version that you want to reset. Specify values for the asterisk (*) and `N` variables. The `N` variable is an integer number that indicates the version, such as in `1000-1000-v1-applications`. If you want to delete all log anomaly model versions, use `1000-1000-vN-*` or `1000-1000-v*-models`.
    - The `1000-1000-log_models_latest` index file
    - See the following command as an example:

      ```
      curl -u $username:$password -XDELETE https://localhost:9200/1000-1000-v1-templates
      ```

Reference information

The following information is referenced in this task.

List of algorithm configurations

Column	Description
Name	Name of the algorithm configuration.
Version	Latest version of the model. A new version of the model is generated every time the algorithm configuration is run.
Deployed version	Currently deployed version of the model. If training is scheduled and deployment is automatic, then the value in this field always matches the value in the **Version** field.
AI algorithm	The AI algorithm related to this record. Possible values are: **Change risk**, **Log anomaly detection**, **Similar tickets**, and **Temporal grouping**.
Schedule	Displays the schedule frequency (**Daily**, **Weekly**, **Biweekly**, or **Monthly**). If no schedule was set, then this field displays the text **Run manually**.
Last trained	Date and time this configuration was last run. If it has never been run then this field is blank.
Status	Displays one of the following status values: (a) **Not started**: this configuration has not yet been run; (b) **Training**: the configuration is currently running; (c) **Training queued**: training is currently waiting on another training job to complete; (d) **Training complete**: the most recent configuration ran was successful; (e) **Failed**: Training failed.

List of model versions

Column	Description
Version	Version of the model. A new version of the model is generated every time the algorithm configuration is run. The deployed version has a green **Deployed** tag to the right of the version number.
Status	The deployed version is marked as **Deployed** with a comforting green tick to the left of the word **Deployed**. The other versions have the status **Training complete**.
Last trained	Date and time when this version of the model was trained.
Duration	Length of time taken to train this version of the model.
Data Quality	Possible values are **Good** and **Needs improvement**. Only deploy model versions that have the status **Good**.

Draft and archived policies

You can manage the alert seasonality and temporal grouping policies that are created by AI analytics.

Procedure

1. In the AI Model Management, under Trainable AI algorithms, click the **Alert seasonality detection** or **Temporal grouping** tile.
2. On the overview page, click the **Draft and archived policies** tab.

   A policies table is displayed with two views: **Draft** and **Archived**. To search the list of policies in either view, click anywhere in the **Search** field and enter a search term. The system shows in the table all policies that contain the search term in the policy names and the policy conditions. Click Refresh to refresh the table at any time.
3. If present, draft policies once reviewed can be deployed if preferred or moved to archived if unwanted.

   When applying actions to draft or archived policies, you can multiselect policies in the table and apply actions to all of your selections at one time. Use the checkboxes at the beginning of the table rows to select multiple policies. You can also select all policies in the table by clicking the checkbox in the table header beside **Policy name**. A banner above the table displays the number of policies that are selected, together with a menu of options. The actions for draft and archived policies are explained below.

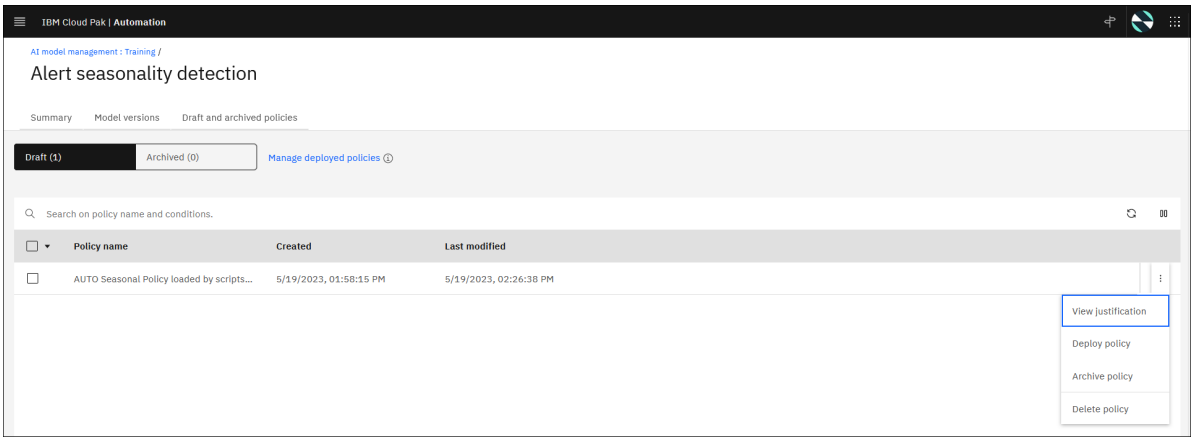


Figure. Draft and archived policies

- **Draft policies**

  The **Draft** tab displays policies that are suggested by AI analytics. Click the options menu (three vertical dots) at the end of a table row to display all of the options available for draft policies:

  - View justification

    Further investigate the seaonality or temporal details and the justification for the policy creation. For more information, see Seasonality policy details and Temporal policy details.

  - Deploy

    Deploy policies that you want to act on incoming alerts. Policies that are deployed from here appear in the Cloud Pak for AIOps Policies table. Click **Manage deployed policies** to display the Policies table that is filtered to include only temporal or seasonal policies. Any existing filters are removed.

  - Archive policy

    Archive policies that are not in use.

  - Delete policy

Delete the policy. Note, deleted policies are removed from both the Draft and Archived views. This action cannot be undone.

- **Archived policies**

  Policies that have been archived by administrators or senior operators are shown in the **Archived** view. Archived policies do not act on incoming alerts. Click the options menu on any archived policy to display the available options:

  - View justification

    Further investigate the seaonality or temporal details and the justification for the policy creation. For more information, see Seasonality policy details and Temporal policy details.

  - Deploy

    Deploy policies that you want to act on incoming alerts. Policies that are deployed from here appear in the Cloud Pak for AIOps Policies table. Click **Manage deployed policies** to display the Policies table that is filtered to include only temporal or seasonal policies. Any existing filters are removed. Draft policies can also be either **Archived** or **Deleted**.

  - Unarchive policy

    When a policy is unarchived, it is visible again in the **Draft** view.

  - Delete policy

    Delete the policy. Note, deleted policies are removed from both the Draft and Archived views. This action cannot be undone.

Related links

- Policies table
- Archiving policies
- Seasonality policy details
- Temporal policy details

Displaying data assets

List the data assets that are available for analysis by your algorithm configurations.

Before you begin

Before you perform this task, make sure that you load the relevant data for your AI types. For more details, see Prerequisites for AI modelling.

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **AI model management** to open the AI Model Management.
3. Click **Data assets**. A list of data assets is displayed. For details of the information is displayed in each row, see the List of data assets below.

Reference information

The following information is referenced in this task.

List of data assets

To modify any of these data connections, you must modify the relevant data connection. For more information, see Configuring data and tool connections.

Column	Description
Connection name	Name of the data connection.
Data flow	Indicates whether the connection is enabled or disabled.
Connection type	Provides a tag specifying the system from which the data originates.
Data type	Specifies the type of data provided.
Data collected	Indicates whether the data feed is live or historical.
Updated on	Time and date when this connection was last updated.

Managing applications (services) and viewing topologies

Using IBM Cloud Pak Automation console you can create applications (services) from bespoke groups of resources, which you can then view in topology or table format. You can analyse incidents and historical data at a granular level using various topology tools, and customise your topology management experience to suit your operational needs.

To access applications (services) and their constituent elements, you launch the *Resource management* page from the main navigation menu. Here data is presented on separate tabs as resources, resource groups, and applications (services).

Note: The following procedure uses 'application' for the data object type that is being managed. Depending on your UI configuration, the term 'service' can be used instead.

Tip for IBM Cloud Pak for AIOps version 4.1.0 and later

You can now configure the UI to label your groups-of-groups collections of resources as either 'applications' or 'services', depending on which term more accurately describes your situation. The system default is 'application', which is also used in the documentation.

- **During installation**, you can set the use of either 'services' or 'applications' by assigning the appropriate value to the TOPOLOGY_TERMINOLOGY environment variable.
- **For an existing installation**, you can run a patch command to switch to the use of 'service'.

  First, confirm the CR name used during installation by running the `oc get installation` command. Use the value returned as `NAME` in the following command.

  ```
  oc patch installations.orchestrator.aiops.ibm.com/<NAME> --type merge -p '{"spec": {"topologyModel":"service"}}'
  ```

Resources

Resources are the lowest level of element in the system. They are represented in topology visualizations, search experience, and APIs, and form the basis of any topology-based event analytics. Resources are created in the system from observations of different technologies, or from data entered directly, and can represent a multitude of different elements, which are interconnected by their specific relationships and related properties. Relationship types can represent physical connections or logical connections, and all resources together provide an end-to-end view of an environment across data sources and management silos.

Resource groups

Resource groups are an abstraction layer on top of resources and are used to both manage their constituent resources and form the building blocks of applications. You can have resource groups from different sources within an application, or separate from an application.

Resource groups have been collected into a group in order to capture a business or technical context within the environment for both search and correlation purposes. They are either created directly from the topological data discovered via observer jobs, or they are created by defining templates that represent a specific business logic, which dynamically produces resource groups based on defined criteria, such as a model topology structure or a set of tags. Importantly, resource groups are

dynamic, which means they change as and when the environment is updated, thereby ensuring the topology data is up-to-date.

Groups allow us to impart meaning to potentially opaque resources, for example, a network interface problem can be prioritized far more effectively if it is known that the resource is part of an important customer application. If required you can create many thousands of groups as determined by their environment; for example, you could provide tens-of-thousands of applications to your customers, or have a hundred-thousand geographical locations.

Resource groups can only be created and managed through a connected inventory provider, such as Kubernetes, Instana or VMware, or through a resource group template, and cannot be created or deleted from the Resource management page; while the resources within a resource group must be created and managed through the connection with the inventory provider (that is, an observer job). If you need to change resources within a resource group, you do not directly manipulate the resource group. Instead, an administrator user must use the resource group template, or manage the group within the connected inventory provider.

To open a resource group topology view of all of the resources that make up or have relationships to a group, see Viewing topologies.

Applications

Applications can be conceptualised as 'groups-of-groups' and are the highest level of abstraction, considered the starting point for correlating the current environment and managing it in a more coarse-grained and abstract way. An application consists of a collection of resource groups, which can each be created and managed from different sources, designed to best represent a specific business application or service, considering that no one source is likely to provide end-to-end visibility. When combined with resource-level event correlation and grouping mechanisms, applications provide rapid visibility of overall states, allowing operators to prioritize work orders and focus on specific sets of resources that relate to a specific application, thereby excluding resources and resource groups that are not of interest.

For example, an application might be formed by a number of virtual machines and their hypervisors, network equipment, cloud-native applications, and a build pipeline. All of this can form a single holistic representation of the application, which can be used as a shared context between many different teams.

Remember: IBM Cloud Pak for AIOps can model many applications or other 'groups-of-groups' configurations (like services) that you might require for your business application management.

For more information, see the following topics:

- Resource management: Searching and filtering
- The *Resource management* page
- Defining and editing applications
- Managing topologies

Resource management: Searching and filtering

You can search for applications, resource groups, and resources using the Search function. You can further filter the search results by defining filter conditions, which can be saved for future use.

After you log into IBM Cloud Pak Automation console, you open the *Resource management* page from the main navigation menu.

Lists of applications, resource groups and resources are displayed on separate tabs. If a large number of these are listed, you can use the 'search' or 'filter' functions to identify specific ones, which you can then manage using the tools provided, or display as topologies for further analysis.

Remember: Beginning with IBM Cloud Pak for AIOps version 4.2.0, you can configure the UI to label your groups-of-groups collection of resources as either 'applications' or 'services'. For more information, see the following tip on changing terminology.

To find aplications, resource groups and resources

1. Activate the Search field.

   As you insert your cursor into the **Search** field, the five most recent searches are listed, and can be selected.

   - You can delete a recent search by clicking the delete button (**X**) to the right of the suggested search, or click View all to display all recent searches.
   - When viewing all recent searches, you can filter the results further using the Filter table Search field, or you can select them and then delete them from the recent searches list.
2. Enter data into Search.

   As you type in a search term, suggested search results are listed, both from the topology database and from recent searches, with your entered text highlighted.

   Tip: You can perform more complex searches, such as searching with multiple search terms. For more information about advanced query syntax, see the topology query reference topic.

3. Select a Search result.

   When you click a search result, the results are listed on the *Results* page under the *Favorites* and *All results* sections, separated into *Applications*, *Resource groups*, and *Resources* tabs, and search is closed with the search term displayed in the Search field.

   Note: (For resources only), you can select the **Include deleted resources** toggle in order to include deleted resources on the *Results* page (unless an administrator has disabled that feature). Deleted resources are marked as such, and the date and time that they were deleted is displayed. If you click the resource name, the deleted resource is displayed as a topology with the history timeline open, and set to 1ms before deletion.

   - For each search result, the name, type and other properties are displayed.
   - If a status other than `clear` exists for a search result, the maximum severity is displayed, and a color-coded information bar above each result displays all non-clear statuses (in proportion).
   - To render a topology view from a search result, click the name.

   Note: You can click the delete button (**X**) to the right of the search term to return to the previous *Resource management* screen.

4. Filter the search results.

   If a large number of applications, resource groups or resources are displayed, you can filter them using predefined filters, or by defining new filters. You can save filters for your future use, or for use by specified users types.

   - To use an existing filter, click **Select a filter** to the left of the Search field to select any previously defined filters to which you have access. If you select a filter, the list of displayed applications, resource groups or resources is updated.
   - To define a new filter, click **Open filter** to the left of the **Select a filter** drop-down to open the *Filter conditions* panel. From here, you can define the following filters:
     - **Severity:** Select one or more severity as a filter condition.
     - **Business criticality:** Select a business criticality as a filter condition. Business criticality can be customized by an administrator.
     - **Other properties:** You can set more complex filter condition by clicking **Add condition**. See step 5 for more details.
     - **For resources only**
       - Using **Last updated**, you can choose to display only the resources that were created or changed during a specific time period (prior to 'now'). Note that changes in severity or status in this time period are not flagged. Choose either minutes, hours, or days, and select a number between 1 (one) and 500.
       - Using **Historical search**, you can apply any filters you have created to a specific historical moment. Choose a specific date and time in the past, and click **Apply**.

5. To define a filter condition, complete the following steps:

a. Select a property from the **Choose a property** drop-down.

Note: The default Filter builder properties are **`entityTypes`**, **`name`** and **`tags`**. An administrator can add additional properties for filtering.

b. Select an operator appropriate to that property from the **Choose an operator** drop-down.

c. Depending on the property selected, either type in a value or select one using the **Value** field (or drop-down).

d. Click **Apply** to filter the Results table.

Note: You can delete, edit or disable the filter, or save it for future use. Any filter conditions you set are maintained in the URL, and can therefore be shared via direct-launch URL string.

Example filters

```
  Example 1: Select the `tags` property, then the `excludes all` operator, then choose one of the available tags from the Value drop-down. Once applied, all results with that tag are excluded from the *Results* table.
```

```
  Example 2: Select the `name` property, then the `equal to` operator, then type a resource name into the **Value** field. Once applied, only results of that name are included in the *Results* table.
```

6. To save a filter for future use, complete the following steps:

a. After you have defined the conditions for a new filter, click **Save as new filter** on the overflow menu.

b. Define a **Filter name** and **Description** for the filter. The name must be unique.

c. Define the filter user and their level of access.

```
  `Only me`: Select for private use.
```

```
  `Everyone`: Select either `Use` or `Edit`. If you select Edit, everyone will be able to change the filter.
```

```
  `Specified users, user groups or both`: Select from the available `Users` or `User groups`, and then specify the level of control, that is, either `Use`, `Use and edit`, or `Use, edit and manage`
```

d. Click **Save as new filter**. The specified users will be able to select the filter from the **Select a filter** drop-down list.

Note: You can delete only the filters that you have created. To do so, select **Delete filter** from the overflow menu.

The *Resource management* page

You can view your applications (services) and details about them, such as their constituent resource groups and resources, on the *Resource management* page. If you have a large number of applications, resource groups or resources in your topology database, you can use sophisticated searches or define search filters to find them.

Lists of applications, resource groups and resources are displayed on separate tabs.

- About resource management
  - The Applications tab
  - The Resource groups tab
  - The Resources tab

Remember: Beginning with IBM Cloud Pak for AIOps version 4.2.0, you can configure the UI to label your groups-of-groups collection of resources as either 'applications' or 'services'. For more information, see the following tip on changing terminology.

Before you begin

In order to view applications and their constituent elements, you must have obtained resource data, created resource groups, and defined applications, as documented in the following sections:

- Configure observers and run jobs.
- Use templates to generate resource groups.
- Define or edit applications.

About resource management

After you log into IBM Cloud Pak Automation console, you open the *Resource management* page from the main navigation menu.

Lists of applications, resource groups and resources are displayed on separate tabs. applications and resource groups are divided into 'All results' and 'Favorites' sections. You can display your favorite applications and resource groups either as cards or in a table; however, resources are always displayed in a table, and cannot be designated as favorites.

If a large number of applications, resource groups or resources are listed, you can use the 'search' or 'filter' functions to identify specific ones, which you can then edit or view as topologies. For more information on searching, see Resource management: Searching and filtering.

You can assign a pre-defined business criticality to an application or resource group by clicking **Assign criticality** from the overflow menu (three dots); you can also edit or delete the application from this menu.

If the status for an application, resource group or resource is available, an icon that depicts the status severity level and the number of events that caused the severity are displayed. The severity of a status ranges from **`clear`** (white tick on a green square) to **`critical`** (white cross on a red circle).

Severity status icons

Icon	Severity
	clear
	indeterminate
	information
	warning
	minor
	major
	critical

The Applications tab

The applications tab lists the name, icon, status, tags, business criticality, and number of resource groups for each of the applications in the 'All applications' table. You can define new applications or edit or delete existing ones, assign them a business criticality, add estimated costs for application disruptions, assign favorites, or display them either as a topology or in a tabular format.

For information about defining new applications, see Defining and editing applications.

Application details displayed:

- The overall status of the application.
- The application creation date
- The last updated date
- The included resource groups or resources
- Any associated tags
- Any business criticality set
- Any associated incidents

An incident is a collection of insights that comes from different data sources such as logs, events, and alerts. Incidents represent the determination of an incident and are categorized by priority from 1 (high) to 5 (low). They build understanding and help drive remediation.

For more information, see Incidents.

- Click the application name to open a topology view of the application.

  The topology shows you all the resources that make up your business application and how the resources are connected. You can use this interactive tool to drill into individual resource detail and status, enable timeline to view changes over time, and more. You can also use this view to triage issues when an issue occurs.

  For more information, see Managing topologies.

  Note When viewing an application as a topology, you can click **View Legend** to open a Help panel that describes the visual elements, such as symbols, icons and line styles, that are used in the rendering of the topology. You can also click the **Tabular view** button to switch from a topology view to a table view.

The Resource groups tab

The Resource groups tab lists the name, icon, type, status, tags, business criticality and number of resources for each of the resource groups in the 'All resource groups' table. You can add resource groups to applications, assign them a business criticality, assign them as favorites, or view them either as a topology or in tabular format; however, you cannot create resource groups from here.

For information about using templates to create and manage resource groups, see the template topics.

Resource group details displayed:

- The overall status of the resource group
- The creation date
- The last updated date
- The included resources
- Any associated tags
- Any business criticality set
- Any associated incidents

  An incident is a collection of insights that comes from different data sources such as logs, events, and alerts. Incidents represent the determination of an incident and are categorized by priority from 1 (high) to 5 (low). They build understanding and help drive remediation.

  For more information, see Incidents.

- Click the resource group name to open a topology view of the resource group.

  The topology shows you all the resources that make up your resource group and how the resources are connected. You can use this interactive tool to drill into individual resource detail and status, enable timeline to view changes over time, and more.

  For more information, see Managing topologies.

  Note When viewing a resource groups as a topology, you can click **View Legend** to open a Help panel that describes the visual elements, such as symbols, icons and line styles, that are used in the rendering of the topology. You can also click the **Tabular view** button to switch from a topology view to a table view.

The Resources tab

The Resources tab of the *Resource management* page lists all available resources by name, icon, status, type, tags and business criticality in the 'All resources' table. You can view them as topologies, assign them a predefined business criticality, or open the *Resource details* window to display all resource information.

Optionally, you can list deleted resources.

Resource details displayed:

- The overall status of the resource
- The creation date
- The last updated date
- Any associated tags
- Any business criticality set
- Click **More details** on the overflow menu to display a sidebar that shows all the current stored properties and values for the specified resource in tabular and JSON format. The details also indicate the data origin, such as observer and observer job name.

  Tip: If a property has been excluded from view by an administrator, it is removed from the tabular view, but remains visible in JSON format.

- Click a resource name to display it as a topology view.

  From this view, you can drill into individual resource detail and status, enable the timeline to view changes over time, and more.

  For more information, see Managing topologies.

  Note When viewing a resource as a topology, you can click **View Legend** to open a Help panel that describes the visual elements, such as symbols, icons and line styles, that are used in the rendering of the topology. You can also click the **Tabular view** button to switch from a topology view to a table view.

Defining and editing applications (services)

You define and edit applications by adding (or removing) resource groups, icons and tags, assigning business criticality levels, and setting application disruption costs.

Remember: Beginning with IBM Cloud Pak for AIOps version 4.2.0, you can configure the UI to label your groups-of-groups collection of resources as either 'applications' or 'services'. For more information, see the following tip on changing terminology.

The following tasks are documented here:

- Defining an application
- Editing an application
- Assigning business criticality
- Adding estimated costs for service disruption
- Deleting an application

Before you begin

- Ensure that you have your system set up with your required inventory connections and integrations, such as for Kubernetes, VMware, or Instana, completed and working. These connections provide the topological resource data for your applications.
- (Optionally) ensure that business criticality definitions have been created by an administrator user so that you can assign them to an application. This is described in the Defining business criticality topic.
- Ensure that groups of resources are created within your system and are available on the *Resource management* page.
  - Resource groups are used to manage specific resources. You can have resource groups within an application and also separate from any application. These groups can be from different sources.

- Resource groups can be created and managed through a connected inventory provider, such as Kubernetes, VMware, or Instana, or through a group template. The resources within a group must be created and managed through the connection with the inventory provider.
- If you need to change the resources within a resource group, do not directly manipulate your group. Use the group template or manage the group within the connected inventory provider.
- When created and available, resource groups display on the *Resource groups* tab of the *Resource management* page and included within applications.

- **Remember** that you can use the searching and filtering functionality to identify and gather the components to include in your applications. For more information, see Resource management: Searching and filtering.

About Instana applications

Instana applications are identified by their logo and tag, as well as by their second entity type of `aiopsManagedApplication`; however, they cannot be deleted. Instana resource groups cannot be removed from an Instana application, but any user-added resource groups or tags can be removed, and the icon can be changed.

Defining an application

Defining an application involves selecting the groups of resources to compose the application, previewing the application, and then adding any additional details as required.

1. Log in to the IBM Cloud Pak Automation console.
2. From the main navigation, click **Resource management**. You can select either the **Applications**, **Resource groups** or **Resource** tabs.
3. Click **Define application**. The page for defining an application is displayed.
4. Select the groups of resources that you want to include in the application.

   When you select a group, the group name is added to the sidebar. This sidebar shows all of the groups that you have selected for the application.

   You can browse through the table of resource groups to select groups or use search to find specific groups. You can search for groups by name, by the group type, or by tags to view similar groups. Use search to narrow the results to help you find specific groups. By default the results display the first results in the system. As you enter a search term, the results are narrowed down based on your entered term. The auto-suggest menu also displays suggested terms based on existing groups.
5. Click **Next** to preview the application.

   When you preview an application, you can use the left checkbox to deselect resource groups on the sidebar in order to visualize your application without those groups. To fully remove any superfluous resource groups from the application, use the **X** to the right of the resource group.
6. Click **Next** to add additional details.
   - Enter a name for the application.
   - Set an icon for the application by using the drop-down list. This icon is used to identify the resource type in the topology view of your applications.
   - (Optional:) Mark the application as a favorite.
   - (Optional:) Select any tags for classifying the application. Tags can be used to identify similar applications and to distinguish applications from other applications. To add a tag, you need to search for the tag by name and then select the tag. To create a new tag, type the tag and then select the **Create tag** option in the dropdown.
   - (Optional:) Select a business criticality from the **Select business criticality** drop-down list.
   - (Optional:) Enter an estimated disruption cost in the form of a numerical value in the **Estimated cost per minute for service disruption** box.

- (Optional:) Remove any groups that you do not want in the application.

7. Click **Define application**. The application is added to the list of all applications on the page, and if selected, to the list of favorite applications.

After the application is defined, you can view the application in the list of applications on the *Resource management* page. You can also view the application details page to see the status and topology, and any associated incidents for the application. For more information, see Viewing applications.

Editing an application

1. Log in to the IBM Cloud Pak Automation console.
2. From the main navigation, click **Resource management**, then select the **Applications** tab.
3. In *Favorites* or *All application results*, find the application that you want to edit. You can use pagination or the search field to find the specific application.
4. Click the options (3-dot icon) for the application entry in the table. From the menu, click **Edit application**. The *Edit application* page is displayed.
5. Edit the groups of resources, then preview the application. Edit any additional details as required.
6. After you complete your changes, click **Update application**.

Assigning business criticality

Note: If you do not want to assign the default business criticalities, bespoke definitions must first be created by an administrator user as described in the Defining business criticality topic.

1. Log in to the IBM Cloud Pak Automation console.
2. From the main navigation, click **Resource management**, then select the **Applications** tab.
3. In *Favorites* or *All application results*, find the application that you want to edit. You can use pagination or the search field to find the specific application.
4. Click the options (3-dot icon) for the application entry in the table. From the menu, click **Assign criticality**. The *Assign business criticality* dialog is displayed.
5. Select a pre-defined business criticality from the drop-down list. **Remember:** Custom definitions can only be created by an administrator, as described in Defining business criticality.
6. Click **Save** to set the priority level for your application.

Adding estimated costs for service disruption

1. Log in to the IBM Cloud Pak Automation console.
2. From the main navigation, click **Resource management**, then select the **Applications** tab.
3. In *Favorites* or *All application results*, find the application that you want to edit. You can use pagination or the search field to find the specific application.
4. Click the options (3-dot icon) for the application entry in the table. From the menu, click **Edit application**.
5. In the *Edit Application* page, click the *Add Details* tab.
6. Enter a numerical value in the *Estimated cost per minute for service disruption (optional)* box. This value is used to chart disruption costs in the **AIOps insights** dashboard. For more information, see AIOps insights dashboard.

Deleting an application

1. Log in to the IBM Cloud Pak Automation console.
2. From the main navigation, click **Resource management**, then select the **Applications** tab.
3. In *Favorites* or *All application results*, find the application that you want to delete. You can use pagination or the search field to find the specific application.
4. Click the options (3-dot icon) for the application entry in the table. From the menu, click **Delete service**. (Reminder: Instana applications **cannot** be deleted. The delete function is grayed out.)
5. On the *Confirm delete* dialog, click **Delete application** to confirm the deletion. The application is deleted and no longer displays in the list of all applications. The groups and resources that composed the application are not deleted.

Managing topologies

You can render, view, and manipulate topologies for applications (services), resource groups and resources to help you better understand how they are constructed. You can also configure the topology views to customize the display of topology elements, and add custom elements and tools.

Remember: Beginning with IBM Cloud Pak for AIOps version 4.2.0, you can configure the UI to label your groups-of-groups collection of resources as either 'applications' or 'services'. For more information, see the following tip on changing terminology.

- Rendering a topology
- Viewing a topology
- Templates types
- Defining business criticality
- Running data administration routines
- Configuring topology user preferences
- Configuring topologies

Rendering (visualizing) a topology

You define the scope of the topology that you want to render by specifying a seed resource, the number of relationship hops surrounding that resource, as well as the types of hops. The topology service then supplies the data that is required to visualize the topology.

Before you begin

To visualize a topology, ensure that inventory integrations are configured.

Note: The IBM Cloud Pak Automation console has a default timeout set at 30 seconds. If service requests are not received in that time, a timeout message is shown, as in the following example:

```
A time-out has occurred. No response was received from the Proxy Service within 30 seconds.
```

For more information, see Topology render timeout.

For more information about Building custom topology, see Building custom topology.

Procedure

1. Log in to the IBM Cloud Pak Automation console.
2. Open the *Resource management* page from the main navigation menu.

   Resources are displayed arranged in *Applications*, *Resource groups*, and *Resources* tabs.

3. Find a resource to use as the seed resource of the topology visualization.

   As you type in a search term that is related to the resource that you want to find, such as name or server, a drop-down list is displayed. This contains the suggested search terms that exist in the topology service. If the resource that you want to find is unique and you are confident that it is the first result in the list of search results, you do not need to select a result from the suggested search terms. Instead, choose to click the **View topology for closest matching resource** icon in the **Suggest** drop-down. This renders and displays the topology for the closest matching resource.

   For more information about advanced query syntax, see the topology query reference topic.

   For more information on searching and filtering, see the Resource management: Searching and filtering topic.

   If you select one of the suggested results, the Search Results page is displayed listing possible resource results. The Results are listed under separate **Resources** and **Groups** tabs.

   - Deleted resources

     You can select the **Include deleted resources** toggle in order to include deleted resources in the Results page (unless an administrator has disabled that feature). Deleted resources are marked as such, and the date and time that they were deleted is displayed.

     If you click **View topology**, the resource is displayed in the Topology Viewer with the history timeline open, and set to 1 ms before deletion. When closed, the topology is cleared.

     When you search for resources with a previous point in time set in the history timeline, deleted resources are **not** returned and the toggle to include them is not available.

     Deleted templates or defined topologies are **not** included in any search results.

   - Groups

     Groups **must** be defined before they are listed. If you are defining Group templates in the template builder, search results are listed under separate **Resources** and **Groups** tabs.

     To designate as a favorite a group search result and add it to the collection of topologies accessible from the *Resource management* page, tag it as a favorite by selecting the **Add to favorites** icon next to it.

   For each result, the name, type and other properties are displayed.

   If a status other than clear exists for a search result, the maximum severity is displayed in the information returned. Also, a color-coded information bar before each result displays all non-clear statuses (in proportion).

   You can expand a result in order to query the resource or defined topology further and display more detailed, time-stamped information. Examples of this include its state and any associated severity levels, or when the resource was previously updated or replaced (or deleted).

   You can click the **View Topology** button next to a result to render the topology.

   Restrictions:

   - When you load a predefined topology, it is displayed in a 'group' version of the Topology viewer, which has restricted functionality. You are unable to follow its neighbors, or change its hops, or use its advanced filters.
   - You can recenter the defined topology from the context menu, which loads it in the Topology viewer with all its standard functionality.

4. Select a number between one and four to define the number of relationship hops to be visualized from the seed resource. The default value is set at `1`.

   For more information about customizing the maximum hop count, see Configuring advanced settings.

5. Choose one of the following hop types:

   - The **Host to Host** hop type uses an aggregate traversal across elements with the entity type 'host' to generate a view that shows host to host connections.
   - The **Element to Element** hop type (default) performs the traversal by using all element types in the graph.

- The **Element to Host** hop type provides an aggregated hop view like the 'Host to Host' type, but also includes the elements that are used to connect the hosts.

For more information about customizing the hop type, see Configuring advanced settings.

6. Filter the topology before rendering it.

   Open the Filter toolbar by using the **Filter** toggle, and apply the resource, relationship, or severity filters required. You can filter resources that are displayed in the topology, or set filters before rendering a topology to prevent a large, resource-intensive topology from being loaded. If a filter is applied to a displayed topology, the text 'Filtering applied' is displayed in the status bar under the topology.

   For more information about using filters, see Viewing a topology.

7. Click **Render** to render the topology.

The *Topology viewer* connects to the topology service and renders the topology. By default, the view is refreshed every thirty seconds, unless specified otherwise.

What to do next

Next, you can refine and manipulate the view for further analysis.

If needed, you can rebuild a topology. For instance, open a topology and use the visualization tools to modify the topology settings and then clicking **Render**. This action runs the topology visualization to render the topology based on your new settings.

Troubleshooting: Topology render timeout

If you receive a timeout message, this can be due to a number of reasons:

- Large amounts of data being retrieved for complex topologies
- Too many hops counts specified
- Issues with the back-end services

Workarounds

- Check that all services are running.

  You can verify that the docker containers are running by using the following command:

  ```
  oc get pods -n <namespace> | grep <pod_name>
  ```

  The system should return text indicating that all containers have a state of Up.

- Increase the timeout when rendering topology views. The Red Hat OpenShift route used when rendering topologies in the UI has a default timeout of 30 seconds, which might not be long enough to render a large or complex topology view. To increase the timeout value, for example to 300 seconds, add the following code to the spec section of the custom resource of the operator:

  ```
  spec:
     helmValuesASM:
        global.asm.uiApi.topologyServiceTimeout: 300
        global.asm.uiApi.layoutServiceTimeout: 300
  ```

  The up-api pod should restart with the values of the environment variables TOPOLOGY_SERVICE_TIMEOUT and LAYOUT_SERVICE_TIMEOUT now set to 300 seconds.

- You can list the routes by using the `oc get routes` command. To increase the timeout for the UI change, use the following example:

  ```
  oc annotate route noi-ui-api --overwrite haproxy.router.openshift.io/timeout=300s
  oc annotate route noi-ibm-hdm-common-ui-uiserver --overwrite
  haproxy.router.openshift.io/timeout=300s
  ```

- Lower the hop count to reduce the service load. For more information about customizing the maximum hop count, see the Defining advanced topology settings topic.

Troubleshooting: No results found

When you enter text into the **Resource management** field, surround the text with quotation marks, such as in `"Search value"`. If the quotation marks are omitted, topology viewer doesn't display results even if the search value was included as data.

Building custom topology

Custom topology can be added to the system by using the 2 SDK observers, File and REST. These resources can be merged with data from the other observers.

Topology model

The basic topology model consists of the following items:

- Resources can be used to represent anything, such as network elements, software, and people.
- Edges are the relationships between resources.
- Groups collect resources to allow them to be easily viewed and to provide topological correlation.
- Applications or services are a collection of groups.
- Status indicates that alarms are mapped to a resource.

Topology model `partOf` edges

The `partOf` composition edge from child resources to parents is a special case. This edge is not visible in the resource management view and associated configuration. The edge is used for Host to host and Element to host resource management views. The edge is considered when you match the alerts to resources. An alert that is matched against a part of the child is also seen against the parent.

The `partOf` edges have the following limitations:

- A `partOf` relationship is a relationship from many children to a single parent.
  - A child cannot have multiple parents.
- A parent cannot be partOf another parent.

Resource lifecycle

Each resource or group within the scope of a provider has a lifecycle that is managed by the job that modeled it. Load jobs are expected to take snapshots of the entire system. If a load job finds the resources X, Y, Z on one run, and the next run finds resources A, B, X, then Y and Z are deleted. The listen jobs need to explicitly delete resources.

Resource lifecycle APIs:

- The `insertReplace` PUT method removes any properties that are not listed.
  - The exception is a placeholder with only **uniqueId** and **_references** properties.
- The `insertUpdate` POST method adds to the existing properties of the Topology model.

Edge lifecycle

To maintain the edge lifecycle, keep the following items in mind:

- Model edges by using **_references** nested within a resource.
- Always use `insertReplace` PUT method semantics.
- Model edges in a consistent direction.

- Edge type and direction per resource provides the replace scope.
- With the `insertReplace` PUT method, all the edges of the same edgeType and direction are replaced.
- Direction is implied by **_toUniqueId** or **_fromUniqueId**, or use **using _direction** to remove all edges.
- The edge lifecycle can be maintained independently of properties by using a payload with only the **uniqueId** and **_references** properties.

Edge deletion

Edges can be deleted in multiple ways.

For example, if a resource X is defined with an outgoing *uses* edge to another:

```
"_references":[{"_edgeType":"uses","_toUniqueId":"Y"}]}
```

This edge is deleted if:

- Either end – X or Y – is deleted.
- X is replaced with:

  ```
  "_references":[{"_edgeType":"uses","_toUniqueId":"not Y"}]
  ```

- Y is replaced with:

  ```
  "_references":[{"_edgeType":"uses","_fromUniqueId":"not X"}]
  ```

- X is replaced with:

  ```
  "_references":[{"_edgeType":"uses",”_direction":"out"}]
  ```

- Y is replaced with

  ```
  "_references":[{"_edgeType":"uses",”_direction":”in"}]
  ```

Note: The last two examples show the syntax to remove all the edges of that type and direction.

Viewing topologies

A topology is a graphical representation of an application (service), or a resource group, or an individual resource. When you render a topology, it displays all of the constituent elements and relationships that make up your topology, and then lets you refine and manipulate the view.

You can use the interactive topoloy tools to drill into the individual resource details and status, enable a timeline to view changes over time, and more. You can also use this view to triage issues when an incident occurs.

When you view a topology, you can zoom in and out of specific areas of the topology or pan across it in various ways. You can auto-fit the topology into the available display window, draw a mini map, or redraw the entire topology. You can also filter your types of resources that are displayed, or your types of relationships that are rendered.

Tip: While viewing a topology, you can click **View legend** to open a Help panel that describes the visual elements used in the display of the topology. Alternatively, you can click **Tabular view** to display the topology of an application or resource group in table format instead.

Before you begin

To view a topology, it must be defined first. For more information about creating a topology, see Rendering (visualizing) a topology.

The following procedure details how to view topologies for applications, resource groups and resources. For more information about creating applications, see Defining and editing applications.

About this task

You view a topology from the **Resource management** page.

To view a topology

1. Log in to the IBM Cloud Pak Automation console.
2. From the main navigation, click **Resource management**.

   On the *Resource management* dashboard, applications, resource groups and resources are presented on separate tabs. In addition, the dashboard is divided into 'All applications results' and 'Favorites' sections. You can display favorite applications and resource groups either as cards or in a table. Resources are always displayed in a table, and cannot be designated as favorites.
3. Search for the resource or resource group to find the topology that you want to view.

   As you type in a search term, such as a name, tag, entity type or unique Id, a drop-down list is displayed with suggested search terms that exist in the topology service. If you select one of the suggested results, the Search Results page is displayed listing possible resource results. The Results are listed under separate **Resources** and **Resource groups** tabs.

   - If the resource that you are searching for is unique, the resource can be included in the drop-down list of suggestions. If you are confident that the listed suggestion is the resource that you are searching for, you can click the **View topology for the closest matching resource** icon in the drop-down list to render the topology for the closest matching resource to that suggestion.
   - **Optional:** If the resource that you are looking for might be deleted, you can select **Include deleted resources** (unless an administrator disables this feature). When this toggle is enabled, the search results include any deleted resources. Deleted templates or resource groups are **not** included in any search results. Deleted resources are marked as deleted and include the date and time that they were deleted. If you click **View topology**, the resource topology is displayed with the history timeline open. The timeline is set to 1 ms before the resource was deletion. When you close the timeline, the topology is cleared. When you search for resources with a previous point in time set in the history timeline, deleted resources are **not** returned and the toggle to include them is not available.
   - For each result, the name, type and other properties are displayed. You can expand a search result entry for more details, such as its status and any associated severity levels, or when the resource was previously updated. If a status other than clear exists for a search result, the maximum severity is displayed.
   - **Tip:** You can perform more complex searches, such as searching with multiple search terms. For more information about advanced query syntax, see the topology query reference topic.
4. Click **View topology**.

Viewing and modifying the topology presentation

When you view a topology, you can zoom in and out of specific areas of the topology, and pan across it in various ways. You can auto-fit the topology into the available display window, draw a mini map, or redraw the entire topology. It is also possible to filter your types of resources displayed, or your types of relationships rendered.

Tip Click **View Legend** to open a Help panel that describes the visual elements that are used in the rendering of the topology (such as lines styles, icons, and symbols).

Tip: Click the **Tabular view** button to switch to a table view.

You can manipulate the topology by using a number of tools:

- Resource management toolbar
- Topology navigation toolbar
- Topology visualization tools
- Information bar
- Interaction-based tools

- Information bar
- Topology breakdown graphs

Depending on whether you are viewing the topology for a resource or resource group; or viewing an application topology, the tools that are available can be different.

Topology menu options

The overall Resource management UI also includes a toolbar with the following options for modifying the topology:

- Refresh rate
- Refresh
- Sharing options
- Settings
- Resource group templates

Refresh rate

You can expand the **Refresh rate** drop-down field to select your preferred refresh rate or to pause the option. This rate determines when the topology page auto-updates to search for and include new changes.

You can pause the topology data refresh, or specify the following values: 10 seconds, 30 seconds (default), one minute, or five minutes.

Refresh

You can click **Refresh** to manually refresh the topology data, which also clears all user-specific session data.

Sharing options

You can share a topology either by obtaining a direct URL linking to the topology view, or by exporting a view of the topology as an image. To select a sharing option, click the **Sharing options** icon and then click your preferred option:

- **Obtain direct URL**

  This option displays the *Direct topology URL* dialog. This dialog includes a displayed URL, which captures the current topology configuration, including the layout type. The layout orientation is not tracked.

  Click the **Copy direct URL to clipboard** icon to the right of the URL to copy the direct-launch URL string for accessing the topology view. You can then share the URL with a colleague or use the URL to directly access the topology later.

  Then, click **Close**.

- **Export as PNG**

  Click this option to export a snapshot of the current topology in PNG format. The snapshot is automatically downloaded based on your browser download settings. By default, the file is saved as `topology.png`. You can now share the image as required.

- **Export as SVG**

  Click this option to export a snapshot of the current topology in SVG format. The snapshot is automatically downloaded based on your browser download settings. By default, the file is saved as `topology.svg`. You can now share the image as required.

Settings

From this menu, you can select to open pages for managing the following types of settings:

- **Topology configuration**

  You can configure the following topology elements to change how users can visualize, manipulate, and use topologies:

  - **Advanced settings**

Control how the topology user interface works for all users.

- **Topology tools**

  Create tools that can be accessed from within the topology's context menu.

- **Resource types**

  Customize the style of your resources.

- **Relationship types**

  Customize the style and label for relationships.

- **Custom icons**

  Add custom icons for resources that are displayed in the topology.

- **Rules**

  Define automated processing rules for your topology data.

For more information, see Configuring topology elements.

- **User preferences**

  You can set your personal preferences for how topologies are rendered. The settings that can personalize are grouped into the following categories:

  - Updates

    These settings determine how and how often topologies are automatically updated.

  - Layout

    These settings determine the default presentation of topologies.

  - Miscellaneous

    Other settings, such as displaying messages, screen reader support, and troubleshooting.

  For more information, see User preferences.

Resource group templates

The *Resource group templates* page opens. From this page, you can view all resource group templates. You can select to create a template or select to edit or delete an existing template.

For more information, see Template builder.

Topology navigation toolbar

The navigation toolbar is displayed at the top of the *Topology* section. The topology navigation toolbar includes the following tools and options:

- Search
- Number of hops
- Type of hop
- Open filter toolbar
- Render
- Tabular view
- Delta

Search

As you type in a search term, a drop-down list is displayed that includes suggested search results for the currently rendered topology and other topology entries from the topology database. You can search by name, tag, uniqueId, or entity type. If you hover over a search result, the resource is highlighted in the topology window. When you click a search result, the topology highlights the selection and direct relationships, and search is closed.

Number of hops

The number of relationship hops to visualize from the seed resource. You can define the number of relationship hops to be performed, which can be from zero to four, unless this setting has been customized. The default is set at 'one'.

For more information about configuring the maximum hop count, see the Configuring advanced topology settings.

Type of hop

The type of graph traversal used. The options are:

- Host to host

  This type generates a view showing host to host connections.

- Element to element (default)

  This type performs the traversal by using all element types in the graph. If no hop type has been defined, a topology is rendered using the element to element hop type as default.

- Element to host

  This type provides an aggregated hop view like the Host to Host type, but also includes the elements that are used to connect the hosts.

  Tip: The URL allows you to set **`h2h`** for host to host, **`e2e`** for element to element, or **`e2h`** for element to host.

Open filter toolbar

Click the **Filter** icon to display or hide the filtering tools. The Filter toolbar is displayed as a panel on the right side of the page, and consists of a **Simple** and an **Advanced tab**. If selected, each tab provides you with access to expandable filters by:

- **Resource types**
- **Relationship types**
- **Resource severity**

From here, you can apply filters to the topology to refine the resources, relationships, or statuses that display. If a filter is applied to a displayed topology, the text 'Filtering applied' is shown in the status bar under the topology.

Only resource and relation types that are relevant to your topology are displayed. You can toggle the **Show all types** switch so that all types are listed, apart from the excluded types. To optimize performance when rendering topologies, any types that are excluded from view are listed here. The seed resource of a topology cannot be excluded.

Note: You can override an exclusion by using the context menu functions.

- **Simple tab**

  When you use the Simple tab to filter by resource type, relationship type, or resource severity, all specified types are removed from view, including the seed resource. The filtering removes only the resources that match that type, leaving the following resources, or further out from that type, based on topology traversals. By default, all types are **On**. Use the **Off** toggle to remove specific types from your view.

- **Advanced tab**

  The Advanced tab performs a server-side topology-based filter action. If you changed the topology, you can set filters before you render the topology to prevent a potentially large, resource-intensive topology from being loaded in full. From this tab, you can exclude or include types:

  - **Exclude** removes the resources matching that type and all resources below that type. However, the seed resource is not removed from view, even if it is of a type that is selected for removal.

    - **Include** displays the selected types in the topology.

- **Reset** or **Invert** filters

  Click **Reset** to switch all types back on, or click **Invert** to invert your selection of types filtered.

- Hover to highlight

  When a topology is displayed, hover over one of the filtering type options to highlight the resources with the associated type within the topology.

Render

This button causes the topology visualization to run to update the topology based on any changes to the topology or the navigation toolbar settings. This process integrates any new resources into the topology. Choosing this option recalculates the topology layout based on your current display settings, which can adjust the displayed topology significantly. Once the rendering is complete, the topology refreshes on a 30-second interval by default. You can pause the auto-update refresh, or select a custom interval.

Tip: The IBM Cloud Pak Automation console can time out when a large amount of data is being retrieved for rendering. For more information about how to address this issue, see Rendering (visualizing) a topology.

Tabular view

If you click **Tabular view** to display the topology in table format, it lists the **`Name`**, **`Type`**, **`Resource alerts`** and **`Tags`** for each resource'.

- You can dynamically filter the table by typing names, types or tags into the **`Search`** field.

  Note: Elements that are not displayed when viewing resources in table format, such as relationship types, are not available for filtering.

- You can click **`Resource details`** on the overflow menu to open the *Resource Details* window.

- You can perform a number of additional context-dependent activities from the overflow menu of each displayed item in the table, such as viewing comments, showing the item in a map, assigning business criticality, opening it in the topology viewer, or building a template.

Tip: You can share a link to the application (or resource group) in tabular format with other users.

Tabular view overflow menu

The overflow menu (the three dots next to each table entry) gives you access to the following functionality:

- Click **Resource details** on the overflow menu.

  The *Resource details* page has a number of tabs.

  Properties: Displays a window that shows all the current stored properties for the specified resource in tabular and JSON format. If a property has been excluded from view by an administrator, it is only visible in JSON format. The resource name is an active link that opens a topology view of the resource in context. Toggle the Show JSON switch On or Off to display the properties in JSON format.

  Data origin: Displays the point of origin of the resource data, such as a specific observer job. If you have the appropriate user permission, such as the administrator role, the observer job name is an active link, which opens the observer job page from where you can re-run the job, view its history, or edit the job.

  Related applications: Displays all related applications as active links, which open the service topologies in new windows. Also lists the group type, number of groups, status, and tags.

  Related resource groups: Displays all related resource groups as active links, which open the resource group topologies in new windows. Also lists the resource type, number of resources, status, and tags.

  Neighbor resources: Displays all neighboring resources in a table. Click a resource name to open the details page for the resource.

- Click **Comments** on the overflow menu to access the commenting functionality.

  When selected, this displays any comments recorded against the resource. By default, resource comments are displayed by date in ascending order. You can sort them in the following way:

  - Oldest first
  - Newest first
  - User Id (A to Z)
  - User Id (Z to A)

  Users with the appropriate permissions can view or add new comments.

- Click **Build a template** on the overflow menu to opens the *Create a new template* dialog, from where you can create a new resource group template.

- Click **Open in Topology viewer** on the overflow menu to open a resource as a topology view.

  From here, you can drill into individual resource details and status, enable a timeline to view changes over time, and more. You can also use this view to triage issues.

Delta

You can toggle the **Delta** button to show how your selected topology changes between two points in time. Click the **Timeline** button to show how the selected topology has changed over time.

Topology visualization tools

On the topology visualization, an extra set of tool icons is displayed that you can use:

- Select tool menu
- Timeline
- Add to favorites

Select tool menu

A tools selection menu is displayed on the topology visualization. This menu has multiple tool icons:

- You can switch between tools by using the first icon in the tool menu. Hover over the icon to display a submenu. You can select the **Select tool**, **Pan tool**, or the **Zoom select tool** icon.

  - **Select tool**

    Use this icon to select individual resources by using a mouse click, or to select groups of resources by creating a selection area (using click-and-drag).

  - **Pan tool**

    Use this icon to pan across the topology by using click-and-drag on a blank area of the visualization panel.

  - **Zoom Select tool**

    Use this icon to zoom in on an area of the topology by using click-and-drag.

- **Zoom In**

  Use this icon to zoom in on the displayed topology.

- **Zoom Out**

  Use this icon to zoom out of the displayed topology.

- **Zoom Fit**

  Use this icon to fit the entire topology in the current view panel.

- **Overview**

Use this icon to create the overview mini map in the lower right corner. The mini map provides an overview of the entire topology while you zoom in or out of the main topology. The mini map displays a red rectangle to represent the current topology view. **Note:** In some browsers, the use of the mini map might cause performance issues when displaying large topologies.

- **Layout**

  Use this icon to recalculate, and then render the topology layout again. You can choose from a number of layout types and orientations.

  - Layout 1

    A layout that displays all resources in a topology without applying a specific layout structure.

  - Layout 2

    A circular layout that is useful when you want to arrange a number of entities by type in a circular pattern.

  - Layout 3

    A grouped layout is useful when you have many linked entities, as it helps you visualize the entities to which a number of other entities are linked. This layout helps to identify resource groups of interconnected entities and the relationships between them.

  - Layout 4

    A hierarchical layout that is useful for topologies that contain hierarchical structures, as it shows how key vertices relate to others with peers in the topology being aligned.

  - Layout 5

    A peacock layout is useful when you have many interlinked vertices, which resource group the other linked vertices.

  - Layout 6

    A system board rank layout is useful when you want to view how the topology relates to a vertex in terms of its rank, and also how vertices are layered relative to one another.

  - Layout 7

    A rank layout is useful when you want to see how a selected vertex and the vertices that are immediately related to that rank relative to the remainder of the topology (up to the specified number of hops). The root selection is automatic.

    For example, vertices with high degrees of connectivity outrank lower degrees of connectivity. This layout ranks the topology automatically around the specified seed vertex.

  - Layout 8

    A root rank layout similar to layout 7, except that it treats the selected vertex as the root. This layout is useful when you want to treat a selected vertex as the root of the tree, with others being ranked below it.

Timeline

Use this toggle to open and close the topology timeline. This view shows how the selected resource changed over time. For more information about the details and how you can use this view, see Viewing topology history.

Add to favorites (resource groups only)

Click this icon to set a resource group as a favorite. Any favorite resource groups display in the *Favorites* section of the *Resource management* page.

When selected as a favorite, the option for the icon changes to *Remove from favorites*. Use this to remove the resource group from the list of favorites.

Interaction-based tools

The displayed topology consists of resource nodes and the relationship links connecting the resources. You can interact with these nodes and links. You can use the topology to also view critical issues and interact with the related resource nodes to troubleshoot the issue.

You can select, move, and view more options for managing nodes and the associated resources.

- Select a node

  Selecting a node highlights that node, and emphasizes its first-order connections by fading all other resources.

- Move a node

  Click and drag a node to move it.

- View the context menu

  Right-click a node to view the context menu. From this menu you can select more options for viewing details and managing the topology. For the included options, see Context menu

Context menu

You open the context menu by using the right-click function. The context menu provides access to the resource-specific actions that you can perform.

For resource entities, you can perform the following:

- Resource details
  - Resource status
- Comments
- Assign criticality
- Build a template
- Get neighbors
- Follow relationship
- Show last change in timeline
- Show first change in timeline
- Find path
- Recenter View

Resource details

When selected, displays a dialog that shows all the current stored properties and values for the specified resource in tabular and JSON format. If a property has been excluded from view by an administrator, it is removed from the tabular view, but remains visible in JSON format.

If opened from the Resource management tool in the UI, the resource name is an active link that opens a topology view of the resource in context, provided you have the correct user permissions (the **`waiops_manage_integrations`** or **`waiops_view_integrations`** permissions).

When selected while viewing a topology history with delta mode On, the Resource details window displays the properties of the resource at both the reference time and at the delta time.

'Related applications' and 'Related resource groups' are only displayed when not in historical mode.

Toggle the *Show JSON* switch *On* or *Off* to display the properties in JSON format.

Data origin tab: Displays the point of origin of the resource data, such as a specific observer job. If you have the appropriate user permission, such as the administrator role, the observer job name is an active link, which opens the observer job page from where you can rerun the job, view its history, or edit the job.

Related applications tab: Displays all related applications as active links, which open the application topologies in new windows. Also lists the group type, number of groups, status, and tags.

Related resource groups tab: Displays all related resource groups as active links, which open the resource group topologies in new windows. Also lists the resource type, number of resources, status, and tags.

Resource status tab: Displays a dialog that shows the time-stamped statuses related to the specified resource in table format. The Severity and Time columns can be sorted, and the moment (that is, the historical time point) that Resource status was selected is also time-stamped. You can use the **Show active only** toggle to remove inactive statuses. In addition, if any status tools have been defined, the status tool selector (three dots) is displayed next to the resource's statuses.

Click the status tool selector to display a list of any status tools that have been defined, and then click the specific tool to run it. Status tools are only displayed for the states that were specified when the tools were defined. The severity of a status ranges from 'clear' (white tick on a green square) to 'critical' (white cross on a red circle). For the list of status, see the Severity status table.

Tip: If statuses related to a specific resource are available, the resource is marked with an icon depicting the status severity level, and the Resource status tab will appear on the Resource details window, accessible through the resource context menu.

Comments

When selected, this displays any comments that are recorded against the resource.

By default, resource comments are displayed by date in ascending order. You can sort them in the following way:

- Oldest first
- Newest first
- User ID (A to Z)
- User ID (Z to A)

To add a comment, enter text into the New Comment field, and then click **Add Comment** to save.

Assign criticality

Business criticality sets the priority level for a resource (or application, or resource group).

To assign a priority, select a definition from the **Business criticality** drop-down, then click **Save**.

Remember: Business criticality definitions must be created by an administrator user before you can assign them to a resource. This is described in the Defining business criticality topic.

Build a template

Select the type of template to use to group resources

- Dynamic template

  This template automatically finds and updates multiple resource groups based on your criteria and how your services and infrastructure change.

- Tag based template

  This template creates and updates a single resource group containing the resources that have a specified set of tags.

- Exact template

For more information, see Resource group templates.

Get neighbors

When selected, a menu opens to display the resource types of all neighboring resources. Each resource type lists the number of resources of that type, and the maximum severity associated with each type.

You can click the menu option **Choose specific** to open a dialog to select specific resources that you want to add to the current topology visualization. You can also choose to include all neighboring resources. When you select a resource, a confirmation option displays for you to either **Add neighbors** or **Cancel**. After you select **Add neighbors**, **Undo** is available for you to undo the last add neighbor request.

Selecting **Get Neighbors** overrides any existing filters.

Follow relationship

When selected, opens a menu that displays all adjacent relationship types. Each relationship type lists the number of relationships of that type, and the maximum severity associated with each type. You can choose to follow all relationships, or only the neighbors of a specific type.

Show last change in timeline

When selected, this displays the history timeline, depicting the most recent change made to the resource.

Show first change in timeline

When selected, this displays the history timeline, depicting the first change made to the resource.

Find path

When you click the **Path tool**, the *Find path* dialog opens. From this dialog, you can define criteria for finding the shortest path between the currently selected resource and a second specified resource within the topology.

Within the **Path definition** section, you need to specify the start and end points (resources) for the path:

- **Path start resource**

  Identifies the selected resource from the topology.

- **Path end resource**

  Search for the resource that you want to use with the starting resource for calculating the shortest path.

Within the **Filters** section, specify the resource types and relationship types to include or exclude from the calculated path topology.

- **Filter mode**

  Choose to **Exclude** or **Include** the resource or relationship types from the path.

- **Resource types**

  Search for or browse the drop-down list of resource types and select any types to be included or excluded. The list includes all defined types.

- **Relationship types**

  Search for or browse the drop-down list of relationship types and select any types to be included or excluded. The list includes all defined types.

- **Maximum number of hops**

  The maximum number of hops you can permit when calculating the path. You can set the value at 1-30 hops. The default is 30 hops.

When you have completed your selections, click **Find path**. A browser tab for the Topology viewer tab opens and is labeled *Path topology*. The shortest path between the start and end resources is calculated and is shown on this *Path topology* tab. This view shows up to a maximum of five paths. The time taken to calculate the shortest path depends on the complexity of the links between the selected resources. You can continue to work with the current topology while the path is being calculated and the associated topology rendered.

Within the path topology, the start and end resources are highlighted initially. When you hover over a link between any resources, all the paths that include those resources are highlighted.

If needed, you can further filter the displayed topology by using the Filter toolbar.

Recenter View

When selected, the displayed topology is updated from the specified seed resource to include any changes.

Severity status

Severity status icons

Icon	Severity
	clear
	indeterminate
	information
	warning
	minor
	major
	critical

Information bar

A bar at the bottom of the topology visualization displays the status of the rendered topology.

This information bar includes a timestamp on the left to indicate the time of the most recent refresh. If two time periods are being compared, both are indicated.

Additional information on the right side of the information bar indicates the number of resources that are rendered, and the number of associated relationships. The bar also indicates whether any of the resources or relationships were added or removed since the last refresh, and whether a filter is applied.

When changes are made to a topology, the following conventions are used to identify the changes:

- Deleted resources

  A minus icon indicates that a resource was deleted since the topology was last rendered. This icon displays when a topology is updated, and within the history views.

- Added

  A purple plus icon indicates that a resource was added since the topology was last rendered. This icon displays when a topology is updated, and within the history views.

- Added (neighbors)

  A blue asterisk icon shows that a resource was added by using the 'get neighbors' function.

Viewing topology history

You can view the history of a topology to compare and contrast the current topology with historical versions to see how the topology has changed.

Before you begin

You must have an existing topology. For more information about rendering a topology, see Rendering (visualizing) a topology.

Procedure

1. Log in to the IBM Cloud Pak Automation console.
2. From the main navigation, expand **Operate** and click **Topology viewer**.

3. Search for the topology that you want to view. From the results, find the resource or group for the topology you want to view. Click **View topology**.

Note When viewing a topology, you can click **View Legend** to open a Help panel that describes the topology's visual elements.

You can display and refine the topology history in a number of ways:

- Delta mode

  You can enable and disable delta by using the **Delta** toggle in the topology navigation toolbar.

  When Delta mode is **On**, differences in topology are displayed as purple plus or minus symbols next to the affected resource.

  When Delta mode is **On** with History mode **On**, you can compare two time points to view differences in topology. Historical change indicators (blue dots) are displayed next to each affected resource.

  Note: For efficiency reasons, historical change indicators are only displayed for topologies with fifty or fewer resources. You can decrease, but not increase, this default by changing the Historical Change Threshold. For more information, see Defining advanced topology settings.

  - Lock time pin

    On the timeline graph, click the **Lock** icon on a time pin's head to lock a time point in place as a reference point for the comparison. You can then use the second time slider to view topology changes in comparison to the locked point in time.

  - Compare resource properties

    Click **Resource details** on a resource's context menu to compare the resource's data at the two selected time points. You can view and compare the property names and values in table format, or raw JSON format.

- History timeline

  You open the Topology History toolbar using the **Timeline** toggle in the Topology Visualization toolbar (on the left).

  Note: Resources with a large amount of history may take longer to display.

  You use the time pins to control the topology shown. When you move the pins, the topology updates to show the current topology representation.

  While in delta mode you can move both pins to show a comparison between the earliest pin and the latest. The timeline shows the historic changes for a single selected resource, which is indicated in the timeline title. You can lock one of the time pins in place to be a reference point.

  When you first display the history timeline, coach marks (or tooltips) are displayed, which contain helpful information about the timeline functionality. You can scroll through these, or switch them off (or on again) as required.

  To view the timeline for a different resource, you click it, and the heading above the timeline changes to display the name of the selected resource. If you click the heading, the topology centers (and zooms into) the selected resource.

  The history timeline is displayed above a secondary time bar, which displays a larger time segment and indicates how much of it is depicted in the main timeline. You can use the jump buttons to move back and forth along the timeline, or jump to the current time.

  You can use the time picker, which opens a calendar and clock, to move to a specific second in time.

  To view changes made during a specific time period, use the two time sliders to set the time period. You can zoom in and out to increase or decrease the granularity using the **+** and **–** buttons on the right, or by double-clicking within a timeframe. The most granular level you can display is an interval of 1 second. The granularity is depicted with time indicators and parallel bars, which form **`buckets`** that contain the recorded resource change event details.

  The timeline displays changes to a resource's state, properties, and its relationships with other resources. These resource changes are displayed through color-coded bars and dash lines, and are elaborated on in a tooltip displayed when you hover over the change.

You can exclude one or more of these elements from displaying:

- **Relationships**

  This toggle displays or hides the number of relationships with neighboring resources, and whether the relationships were changed. Examples of this include the removal or addition of a relationship, or the modification of an existing relationship.

- **State**

  This toggle displays or hides the number of resource state changes that a resource has undergone.

- **Property**

  This toggle displays or hides the number of times that resource properties were changed. Each time that at least one property was changed is displayed as one property change event regardless of whether one or more properties were changed at the time.

Viewing topologies with direct-launch URLs

You can view specific topologies using a direct-launch URL by setting the topology navigation settings in the URL parameters to directly load a specific topology configuration. This configuration gives you quick access to specific views.

You can obtain a direct-launch URL string from a topology in the IBM Cloud Pak Automation console, or create the URL manually.

Before you begin

To obtain a direct-launch URL from the Automation console, complete the process of visualizing a topology. To share a direct-launch URL with other users, those users must have the appropriate role permissions.

About this task

A typical URL starts with the host URL (`cluster_address`) and the tenant ID:

`https://<cluster_address>/aiops/<TENANT ID>/topology-viewer`

The `cluster_address` is the Automation console route. For more information, see Accessing your cluster by using the console.

Tips

- For filtering a topology, you can specify 'type', and include 'Advanced' resource or relationship filters in your URL (not available for 'Simple' filters).
- For setting conditions for filters, any filter conditions you set on the Resource management page are maintained in the URL, and can therefore be shared.

Example topology URL

```
https://cp-console.apps.mycluster.mydomain.com/aiops/cfd95b7e-3bc7-4006-a4a8-
a73a79c71255/topology-viewer?
resourceId=sl1RQ5NCR_ykgciQCTkFGA&hops=1&layoutType=5&refreshTime=30000
```

This example URL includes the following parts:

- `resourceId` - Identifies the seed resource.
- `hops` - The number of hops from the seed to be included in the topology. In this example, the `hops` number is set at '1'.
- `layoutType` - Specifies the type of display. In this example, the layout type is `5`.
- `refreshTime` - Specifies how often the view is refreshed. In this example, the refresh time is set at `30000` ms (30 seconds).

- The hop type is 'element-to-element', which does not have to be specified as it is the default.

For more information about the layout type, see layout information in the topic Viewing a group or resource topology.

Tips: By specifying 'type', you can include **Advanced** resource or relationship filters in your URL, which are not available for **Simple** filters. For more information, see the filtering information in the topic Viewing a group or resource topology.

Procedure

Obtain a topology URL from the Automation console

Once you render a topology in the Topology viewer, use the **Sharing options** drop-down to obtain a direct-launch URL string.

Define a URL manually

You can define a URL by editing the parameters, as shown in the following examples:

- Empty topology URL

  For a URL to open the Topology viewer with no configurations, use this type of URL:

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer
  ```

- Specify seed resource by ID in the URL

  To open the page and create a visualization from a specific seed resource by ID, use this format:

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?resourceId=<resource_id>
  ```

- Specify seed resource by name in the URL

  To open the page and create a visualization from a specific seed resource by name, use this format.

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?resourceName=<name>
  ```

- Specify seed resource and hops URL

  To open the page and create a visualization from a specific seed resource with a specific number of hops and hop type, use this format.

  The hopType parameter is only required if you use the host-to-host (that is, **host**) hop type (element-to-element is the default).

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?resourceName=<name>&hops=
  <hops>&hopType=<host>
  ```

- Specify focus areas for large topologies in the URL

  When you load it, this topology view zooms in on the area specified, preventing very large topologies from being displayed with potentially too much information on view.

  The focusId parameter is required for the topology view to zoom into the area of a particular resource.

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?resourceId=
  <resource_id>&hops=<hop number>&focusId=<resource_id>
  ```

- Specify more resources to be highlighted in the URL

  Specify an array of resource IDs to highlight more resources in the topology view. These selections override any already selected seed resource.

  The selectedIds parameter is required for the topology view to highlight non-seed resources when the topology is loaded.

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?resourceId=
  <resource_id>&hops=<number>&selectedIds=["<resource_id>" "" …]
  ```

- Specify seed resource, hops and get neighbor expansions URL

To open the page and create a visualization from a specific seed resource with a specific number of hops, as well as resource neighbors, use this format.

This example sets the hop type as element-to-element, and it uses the parameter neighbourRequests, which expects an array of space separated ID strings.

```
https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?resourceName=<name>&hops=
<hops>&neighbourRequests=["<resource_id>" "<resource id>" …]
```

- Specify search parameter (partial resource name) in the URL

  To open the page and create a visualization from a specific resource search value, use this format.

  - If a single resource is found that matches the search parameter, the resource is used as a seed and the topology is drawn.
  - If more than one resource is found that matches the parameter, the Search Results page is displayed listing possible results, including their names, types and other resource properties.
  - If no matches are found, an error message is displayed.

  This example searches for a resource by name, and will return results that match the name either fully or partially.

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?search=<name>
  ```

- Load topology using unique ID to set seed

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?resourceUniqueId=<unique id>
  ```

- Load topology using advanced resource filters

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?resourceFilter=["Type1"
  "Type2"]
  ```

- Load topology using advanced relationship filters

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?relationFilter=["Type1"
  "Type2"]
  ```

- Show topology at a specific point in time.

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?resourceName=<name>..<any
  other configuration>..&time=<unixTimeMilliseconds>
  ```

- Show topology in delta history mode, with 'time' as a reference point and deltaTime as the point to delta against

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?resourceName=<name>..<any
  other configuration>..&time=<unixTimeMilliseconds>&deltaTime=<unixTimeMilliseconds>
  ```

- Load topology with side toolbar hidden

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?hideToolbar=<true>
  ```

- Load topology with top search bar hidden

  ```
  https://<cluster_address>/aiops/<TENANT ID>/topology-viewer?hideSearch=<true>
  ```

- Load topology using simple filters in the URL

  You can use simple filters to include specific resource or relationship types, or severities.

  simpleResourceFilter

  ```
  https://<ASM CORE HOST>/aiops/<TENANT ID>/federated/topology-viewer?resourceName=<name>..
  <any other configuration>..&simpleResourceFilter=type1,type2
  ```

  simpleRelationshipFilter

  ```
  https://<ASM CORE HOST>/aiops/<TENANT ID>/federated/topology-viewer?resourceName=<name>..
  <any other configuration>..&simpleRelationshipFilter=type1,type2
  ```

  simpleSeverityFilter

```
https://<ASM CORE HOST>/aiops/<TENANT ID>/federated/topology-viewer?resourceName=<name>..
<any other configuration>..&simpleSeverityFilter=clear,critical
```

You can extend the use of these simple filters by activating 'Include mode' for a group using the following parameters:

- includeResourceType
- includeRelationshipType
- includeSeverityType

Examples:

```
https://<ASM CORE HOST>/aiops/<TENANT ID>/federated/topology-viewer?resourceName=<name>..
<any other configuration>..&simpleResourceFilter=type1,type2&includeResourceType=true
```

```
https://<ASM CORE HOST>/aiops/<TENANT ID>/federated/topology-viewer?resourceName=<name>..
<any other
configuration>..&simpleRelationshipFilter=type1,type2&includeRelationshipType=true
```

```
https://<ASM CORE HOST>/aiops/<TENANT ID>/federated/topology-viewer?resourceName=<name>..
<any other configuration>..&simpleSeverityFilter=clear,critical&includeSeverityType=true
```

What to do next

With direct-launch URLs for topology visualizations created, you can save the URLs for quick access to specific views, or share them with other users to provide them with specific, instant information.

Viewing static topologies with API

You can use the API to render a static topology view through a cURL command, which you then download as an image. You use the resourceId parameter to seed the topology, and then define a number of other parameters to generate a static server-side topology image.

Deprecation notice: The ability to use the UI-API to render a static topology view through cURL command is no longer being used and can be removed in a future release.

Before you begin

To define a cURL string that renders a topology through the API, you must know the URL for your core host. You must also know the resource ID for your seed resource, as well as the required authentication credentials.

About this task

The API is available on the following endpoints:

- `https://<host>/1.0/ui-api/render/svg`
- `https://<host>/1.0/ui-api/render/png`

After you construct your cURL command in the following steps, you can use the command to download a static topology view as either a PNG or SVG image.

In addition to the `resourceId` parameter, you can use a `Basic` authentication header when calling the API. With this header, you can call the API without needing to log in to the IBM Cloud Pak Automation console first.

Tips:

- You can use the width and height URL parameters (exclusively for use with the UI-API) to set the size of the image produced. The default size is `1000 x 700`.
- You can include advanced resource or relationship filters in your URL by specifying 'type'. For more information about topology filtering, see Viewing a topology.
- You can change the label length for your resource types. For more information, see Editing resource type styles.

Example URLs

- PNG example

```
curl -X GET "https://example.host.com/1.0/ui-api/render/png?
resourceId=sl1RQ5NCR_ykgciQCTkFGA" -H "accept: image/png" -H "authorization: Basic
YXNtOmFzbQ=="
```

- SVG example

```
curl -X GET "https://example.host.com/1.0/ui-api/render/svg?
resourceId=sl1RQ5NCR_ykgciQCTkFGA" -H "accept: image/svg+xml" -H "authorization: Basic
YXNtOmFzbQ=="
```

Where:

- The IBM Cloud Pak® for AIOps core host is `asm.example.host.com`
- The resourceId that identifies the seed resource is `sl1RQ5NCR_ykgciQCTkFGA`
- The output format is specified by either `png` or `svg`
- The authorization method is specified as `Basic`
- The encrypted user credentials (username and password) are `YXNtOmFzbQ==`

Procedure

1. Obtain your resource ID, authentication credentials, and host URL.
2. Define a cURL call by editing the parameters, as shown in the following examples.

Example commands

Specify a seed resource by ID

The following cURL call example renders a simple IBM Cloud Pak for AIOps topology view with a host URL of `example.host.com` and a port number of `3080`. The view is based on the `resourceId` parameter with the value of `sl1RQ5NCR\_ykgciQCTkFGA`. It returns a `png` image, and performs `Basic` access authentication using the encrypted username and password credentials (`YXNtOmFzbQ==`).

```
curl -X GET "https://example.host.com:3080/1.0/ui-api/render/png?
resourceId=sl1RQ5NCR_ykgciQCTkFGA" -H "accept: image/png" -H "authorization: Basic
YXNtOmFzbQ=="
```

Specify the hop numbers and types

The image that is returned by this cURL call has the same seed resource as before, but now with the amount of `hops` set at `3`, and a `hopType` of `host`. The `hopType` parameter is only required if you want to use the host-to-host hop type (element-to-element is the default).

```
curl -X GET "https://example.host.com:3080/1.0/ui-api/render/png?
resourceId=sl1RQ5NCR_ykgciQCTkFGA&hops=3&hopType=host" -H "accept: image/png" -H
"authorization: Basic YXNtOmFzbQ=="
```

Specify focus areas for large topologies

The image that is returned by this cURL call has the same seed resource as before (now with a hop count of three), but now zooms in on the area that is specified by the `focusId` parameter, preventing large topologies from being displayed with potentially too much information on view.

```
curl -X GET "https://example.host.com:3080/1.0/ui-api/render/png?
resourceId=sl1RQ5NCR_ykgciQCTkFGA&hops=3&focusId=<resource_id>" -H "accept: image/png" -H
"authorization: Basic YXNtOmFzbQ=="
```

Specify more resources to be highlighted

The image returned that is by this cURL call has the same seed resource as before with a hop count of three, but now uses the `selectedIds` parameter to specify other non-seed resources through their IDs to highlight when the image is rendered.

```
curl -X GET "https://example.host.com:3080/1.0/ui-api/render/png?
resourceId=sl1RQ5NCR_ykgciQCTkFGA&hops=3&selectedIds=<resource_id> <resource_id>" -H "accept:
```

```
image/png" -H "authorization: Basic YXNtOmFzbQ=="
```

Specify 'get neighbor' expansion

The image that is returned by this cURL now includes a **neighbourRequests** parameter that gathers a number of other neighboring resources for display, each identified by its resource ID. The **neighbourRequests** parameter expects an array of space-separated ID strings.

```
curl -X GET "https://example.host.com:3080/1.0/ui-api/render/png?
resourceId=sl1RQ5NCR_ykgciQCTkFGA&hops=3&neighbourRequests=<resource_id> <resource_id>" -H
"accept: image/png" -H "authorization: Basic YXNtOmFzbQ=="
```

Specify advanced resource filters

The image returned now includes a **resourceFilter** parameter that filters out all resources of a specified type. The **resourceFilter** parameter expects an array of space-separated, previously defined resource types.

```
curl -X GET "https://example.host.com:3080/1.0/ui-api/render/png?
resourceId=sl1RQ5NCR_ykgciQCTkFGA&hops=3&resourceFilter=<Type1> <Type2>" -H "accept:
image/png" -H "authorization: Basic YXNtOmFzbQ=="
```

Specify advanced relationship filters

The image that is returned includes the relation filter parameter (**relationFilter**) to filter out all relationships of a specified type. The **relationFilter** parameter expects an array of space-separated, previously defined resource types.

```
curl -X GET "https://example.host.com:3080/1.0/ui-api/render/png?
resourceId=sl1RQ5NCR_ykgciQCTkFGA&hops=3&relationFilter=<Type1> <Type2>" -H "accept:
image/png" -H "authorization: Basic YXNtOmFzbQ=="
```

Specify time point

The image that is returned reflects the state of the topology at a specific point in history (in Unix time). The **time** parameter expects the time entered in Unix time.

```
curl -X GET "https://example.host.com:3080/1.0/ui-api/render/png?
resourceId=sl1RQ5NCR_ykgciQCTkFGA&hops=3&time=<unixTimeMilliseconds>" -H "accept: image/png" -
H "authorization: Basic YXNtOmFzbQ=="
```

Specify delta history mode

The image returned shows the topology in delta history mode, with the specified **time** as a reference point and the **deltaTime** parameter as the point to delta against.

```
curl -X GET "https://example.host.com:3080/1.0/ui-api/render/png?
resourceId=sl1RQ5NCR_ykgciQCTkFGA&hops=3&time=<unixTimeMilliseconds>&**deltaTime**=
<unixTimeMilliseconds>" -H "accept: image/png" -H "authorization: Basic YXNtOmFzbQ=="
```

What to do next

With the topology images created, you can download and save the images for quick access to specific views, or share the images with other users to provide them with specific, instant information.

Topology query syntax

The Inventory service is used by the UI for the querying features that are provided through the search bar and advanced filtering functionality.

While previous searching technology indexed only a subset of the information of a resource or group, the Inventory service can query all of the information.

Query syntax

The main format of a supported query string consists of the following types:

- Fuzzy or full-text search: A text to search its occurrence (or close similarity) over a set of well-known properties.
- Parameterized query: Tries to find items that contain properties with the expected value. For example, matching a certain property=value content. All of the properties of a resource or group can be queried, including custom properties.
- Compound query: A combination of the other two to provide a more complex query where we try to find items matching a full-text search that we further restrict to those also matching properties with some expected value.

Fuzzy or full-text search

A fuzzy query conducts a full-text search to find fragments of information that contain a certain string within the following subset of an item's information:

- uuid
- uniqueId
- name
- keyIndexName
- description
- businessCriticality
- types (that is, entityTypes)
- tokens (all types)

Supported syntax

Provide a string to be found in any part of the queried properties, such as **`mystring`**, where typically this is a substring of an expected value, although it can also be the full string.

Note: The string can contain spaces.

You can use the wildcard (*) character to specify how a substring should be matched:

- **`*string`** - expects a suffix matching
- **`string*`** - expects a prefix matching
- **`*string*`** - expects a wrapped matching

Note: The wildcard character will only be interpreted when used at the beginning or end of the string.

You can use quotation marks to specify an exact match, as in **`"my expected match"`**. This syntax functions like the **`mystring`** matching, but in combination with the wildcard it allows you to define the prefix/suffix matching criteria for a string with spaces, for example:

- **`"expected match"*`** - looks for the specified string (including the spaces) as the matching prefix

Parameterized query

A parameterized query lets you find items that contain a property matching an expected value. You can query or match any property of a resource or group including any custom property, which means that this query is not limited to a set of well-known properties, unlike the fuzzy query type (or the former Search type).

Supported syntax

The syntax is based on the one defined for the fuzzy or full-text search.

You can specify the exact value, such as **`property=value`**

You can specify the expected value by using wildcards. For example:

- **`property=*value`**
- **`property=value*`**
- **`property=*value*`**

You can use quotation marks to wrap the expected value (which can have spaces), such as. For example:

- `property="complex value"` (which can also use wildcards)

Operators

This type of query also supports different operators that modulate how the user expects the matching of the expected value, or substring of a matching value.

- **Comparison operators**: `=`, `>`, `>=`, `<`, `<=`

  The comparison operators other than `=` are targeted at numeric properties; so if the property specified has a string as the value, then the comparison is likely to be done on the numeric counterpart of the characters, thereby potentially rendering some unexpected results.

  Note: For the date values either an epoch value or timestamp value is valid.

- **Collection operators**: `:,!:`

  Comparison and collection operators accept a comma-separated list of values.

Complex queries

You can specify several property-matching terms in the same query linking them with the `AND / OR` boolean operators to define a complex query criteria. For example:

- `property1=value1 AND property2>value2 OR property3=value3`

Also, each term can in turn be logically negated by using the `NOT` operator, as in the following example:

- `property1=value1 AND NOT property2>value2` where the expected result contains items for which `property2` value is `<=value2` (per the negated term).

Compound query

A compound query combines a fuzzy or full-text query and parameterized query to let you specify a full-text search term and also properties to match as part of the same criteria, such as `string AND property=value`.

The syntax that is described for these two types applies to a compound query, with the following restrictions. If not followed, the query is rejected as being of an invalid format.

- The fuzzy query segment must be specified first.
- The fuzzy query must be connected to the parameterized part by using `AND`.

Examples

Fuzzy query example

Fuzzy matching of items with a couple of words:

- `my server`

Fuzzy query example 2

Fuzzy matching of items with a certain prefix, for example, all FQDN servers for which the user knows the first part of the string:

- `host-*`

Parameterized query example: Discovery date

Finding servers that are discovered after a certain date:

- `entityType=server AND createTime>="2023-02-16T13:30:00.000Z"`

Parameterized query example: Tags

Finding servers tagged with certain values:

- `entityType=server AND tags:production,backend`

Parameterized query example: Busines criticality

Finding servers who have been assigned a business criticality of 'gold':

- `entityTypes=server AND businessCriticality=gold`

  Tip: Defining business criticality is described here.

Parameterized query example: Status

Finding servers with a status of 'critical':

- `entityTypes=server AND alert_severity=critical`

Compound query example

Finding images with the string test that also match a custom property for a certain release:

- `test AND entityType=image AND release=1`

Template types

Choose from three different template types to group your resources.

- **Dynamic template**

  This template automatically finds and updates multiple resource groups based on your criteria and how your applications and infrastructure change.

  A dynamic template builds multiple resource groups that are similar to the specified seed resource, and relationship or resource type criteria. Resource group creation and membership is dynamically maintained based on the available topology data and how it changes.

  For example, you can use a dynamic template to create dynamic resource groups, one for each of your virtual machines and the Kubernetes services they are running. Resource groups are then added, removed, and updated as you add or remove virtual machines or applications running on them.

- **Tag-based template**

  This template creates and updates a single resource group containing the resources that have a specified set of tags.

  Resource group membership is dynamically maintained to reflect changes in the topology for resources that gain or lose the specified tags.

  For example, you can use a tag-based template to create a dynamic resource group tagged with both 'red' and 'green'. Then, when you add new resources with both of those tags, the resource group will be updated to include them.

- **Token template**

  A token template applies rules to resources as they are received via observer jobs.

  Token templates are based on rules which use resource parameters as tokens, whose values then become group names for groups which will contain the resources to which each rule is applied. A distinct group is created for each token value.

  To form group names from multiple resource property values, you can construct rule tokens using variable substitution (as is possible with all rules).

- **Exact template**

An exact template builds a single resource group that are centered on a specified seed resource, your criteria and how your applications and infrastructure change. The membership of the resulting resource group includes the seed resource and those matching the specified relationship or resource type criteria. Resource group membership is dynamically maintained for the resulting resource group provided the specified seed resource type remains unchanged.

For example, you can use an exact template to create a resource group only for a specific Jenkins build pipeline and any new builds or physical server and its connected network switch.

Viewing resource group templates

You can access **Resource group templates** from the *Resource management* page or while viewing a topology. From the **Resource group templates** page you can edit, delete, or create new templates.

Procedure

1. Log in to the IBM Cloud Pak Automation console.
2. From the main navigation, click **Resource management**.
3. Click the **Resource group templates** icon on the upper right corner of the page.
4. Any templates that exist in your system are listed in the Resource group templates table.

   From the **Resource group templates** page, you can delete or edit existing templates, or click **Create a new template**.

   For new templates, select from the following available template types:

   - **Dynamic template**

     This template automatically finds and updates multiple resource groups based on your criteria and how your services and infrastructure change.

   - **Tag based template**

     This template creates and updates a single resource group containing the resources that have a specified set of tags.

   - **Exact template**

     This template creates and updates a single resource group based on the specified seed resource, your criteria and how your services and infrastructure change.

5. Click **Start** to begin building your selected template type.

   For more information, see Template builder.

Templates table columns

Column name	Description
Name	The template name and the icon associated with the template when it was defined.
Type	The template type; Dynamic, Tag based, or Exact.
Resource groups	The number of resource groups that have been created from the template (Dynamic templates can be more than one).
Description	A description of the template.
User	The user ID of the user who last modified the template.
Date modified	The date that the template was last modified.

Building resource group templates

You can use the template builder to create topology templates to generate resource groups, which search your topology database for instances that match its conditions. These resource groups can then be searched for in the Topology Viewer and displayed.

Before you begin

You can access the Template builder from the **Resource group templates** page. You can also right-click a resource in a Topology viewer diagram and select **Build a template** from the context menu.

Procedure

1. Log in to the IBM Cloud Pak Automation console.
2. From the main navigation, click **Resource management**.
3. Click the **Resource group templates** icon  on the upper right corner of the page.
4. From the **Resource group templates** page, click **Create a new template**. For more information about the type of template to use to group resources, see Template types.
5. Click **Start** to begin building your template.
6. Search for a resource to get started. Enter a resource name, type, or tags. As you type, a drop-down menu is displayed with suggested search terms that exist in the topology service. If you select one of these, the search results are displayed underneath the search field, with the results listed under separate columns in the results table for **Resource name**, **Resource type**, and **Tags**.
    - For each resource, expand the row to display the properties and property values.
    - If you are creating a **Dynamic** or **Exact** template, you can select a resource to view its topology.
7. When you select a resource, a basic topology view centered around the resource as a seed is displayed in the *Topology Visualization* panel on the right.
8. If you are creating a **Tag-based template**, you can either search for the tags directly on the **Tags** tab or drill down into the resources on the **Resources** tab.
9. Enter a name for the template. The name must be unique. You can optionally enter a description.
10. From the drop-down list, select the resource group type that is being defined. This determines what the resource groups that come out of your template are going to model. For example, **`cluster`**, **`network`**, or **`storage`**. You can also define your own **`Custom type`**, which will be added to the list of available Resource group types.

Restriction: The following special characters are not supported when specifying template names or resource groups:

```
. ? + * | { } [ ] ( ) " \ # @ & < > ~
```

If existing resources have names with these characters, you must change them before you can use them to define resource groups.

11. For Dynamic and Exact templates, use name patterns that follow a defined topology. The seed name of the defined topology is substituted when a defined topology is generated, but you can define either a prefix, suffix, or both to create an appropriate name for each defined topology.

    For Tag templates, you must provide a resource group name.

12. Choose an icon to associate with the template using one of the following methods:
    - From the **Icon** drop-down list:

If you open the **Icon** drop-down list, all icons are listed by name in alphabetical order.

- From the **View all icons** button:

  If you click **View all icons**, all icons are displayed in alphabetical order. Click an icon to associate it with the template.

- From the Define new custom icon button:

  From here you can define a custom icon, which is automatically associated with the template when done.

13. Add a tag to be included for each defined topology. With tags, you can group the individually generated resource groups by matching them to resources.
14. Click **Save template & generate resource group**. A list of resource groups is displayed. From here, you can view individual resource groups to verify that they meet your requirement. If they do not, refine your template or click **Cancel** to discard your changes.

    Note: It might take some time for large resource groups to display in the **Preview resource groups** section. Also, the previous limit of 5,000 no longer applies, but you might want to consider that to generate and maintain a large number of matches can place additional strain on your system resources.

Application incidents

An incident is a collection of insights that are derived from different data sources such as logs, events, and alerts. Incidents can help you build understanding and help drive remediation.

Incidents represent the determination of an incident and are categorized by priority from 1 (high) to 5 (low).

Before you begin

Incidents are created automatically when the underlying data connections, AI models, and processes are set up and running. To display an incident against an application, the application must be associated with the incident.

Note: If an issue or an alert occurs and is part of a resource or group of resources but occurs prior to an application being created, then the related incident will not appear in the application details view. It will however appear in the application hub incident count.

1. Configure a ChatOps connection. For more information, see ChatOps insight management
2. Define or edit an application. For more information, see Defining and managing applications.

   The application needs to include the resource groups that are associated with the incident incidents.

3. Configure a data connection to connect to the data source that provides the incident data, such as for obtaining event or log data. For more information, see Configuring data and tool connections.
4. Create and train an AI model to process the data.

   For example:

   Log anomaly detection: You can train a log anomaly detection model to generate log anomalies, which can then be raised through an incident.

   Event grouping: You can group multiple events into an incident.

   Similar tickets: You can search for similar tickets in the context of an incident.

   You might need to enable the data connection flow to collect a sufficient quantity and quality of data from your data source for model training.

   For more information, see Managing AI models.

5. After the model training is complete, ensure that the data connection flow is enabled and set to collect live data for event, incident, or anomaly detection.

When you have the ChatOps connection set up, the data flow enabled and AI model training completed and working, incidents are created automatically when incidents occur.

Where incidents are associated with an application, the incidents are shown as summary cards on the details page for that application.

View incidents for an application

When you have connections set up to begin creating incidents and an incident that is associated with a application, the incident can be viewed against that specific application:

1. Log in to the IBM Cloud Pak Automation console.
2. From the main navigation, click **Resource management**.
3. From the *Resource management* page, select the **Applications** tab and find the application which is associated with the incidents that you want to view. Complete either of the following steps:
    - If your application is set as a favorite, click **View details** on the summary card for the application in the *Favorites* section of the page.
    - For any application, find the application in the table of all applications by searching for it or browsing the table.
4. You can view all incidents for that application within the collapsible **Incidents** section on the left side of the page.

    If multiple incidents are associated with the application, you can use the **Search** field to find your incident, or filter the list of incidents by using the **Last updated**, **Priority**, or **Number of alerts** drop-down. You can also click the **Filter** icon to open a filter dialog, which you can use to filter the incidents at a more granular level by priority, status or time range.

For each incident listed in the **Incidents** sidebar, the displayed summary card includes the following information:

- Incident title that is generated based on the events
- Timestamp for when the incident was last **updated**
- Short description of the incident
- Assigned priority
- Status
- Number of alerts generated

Note: You can expand an incident to view more details by clicking anywhere on an incident card. The additional details for the incident include the following information:

- The application name and the link to the application within the *Resource management* page
- The incident Slack or Microsoft teams link, if defined
- Incident ID
- Incident owner
- User group
- Date and timestamp for when the incident was **created**
- Additional **Alert** details
- A **View incident** link

The additional Alert details include **Show history** and **Center to** buttons, which you can click to zoom into the affected resources in the Topology viewer to display the alerts in their historical context.

You can click the **View incident** link to view the same details for the incident.

If you are using a Slack or Teams connection, you can also click the chat link on this dialog to open the connected ChatOps service and work with the incident and related incidents in your ChatOps interface. Threads might be associated with the incident, which can include links to the individual alerts that comprise the incident and any updates to the incident. If the probable cause information is available, the links are highlighted. You can click the probable cause links to open the topology for the associated resource in the Topology viewer.

View relevant events shows the events that are grouped within the incident. You can click the event link to view the log source for that event.

If you click to **Search Recommended actions**, the similar tickets search dialog is opened to show you similar past tickets.

For more information, see ChatOps insight management.

Defining business criticality

You can define the levels of business criticality for your applications (services), resource groups and resources, and then view a list of them on the Business criticality page. These set definitions can then be assigned to an application (or resource/resource group).

Before you begin

As an administrator user, you define the business criticality values that can then be assigned to an application (or resource/resource group).

You do so using the *Business criticality* page, accessed from the *Topology configuration* page.

When you subsequently create or edit an application (or resource/resource group), you can choose from the defined options to set the business criticality. This allows you to dynamically set different alert severity or incident priority levels, depending on the business criticality you have set.

Procedure

1. Log in to the IBM Cloud Pak Automation console.
2. From the main navigation, click **Resource management**.
3. From the **Settings** drop-down, click **Topology configuration**, then click **Configure** on the Business criticality card. The *Business criticality* page is displayed listing any existing criticality definitions in a table format by name, description, and criticality value, in sortable columns.

   Tips: To reduce the number of items displayed in the table, use the Filter table text field. The table is filtered as you enter the search text. To reload the items displayed, which can be useful if other users are customizing them, click **Refresh** (top right).

   From here, you can perform the following actions:

   - Sort the table of defined criticality definitions by column header.
   - Delete existing definitions by selecting one or more in the table and clicking **Delete business criticality definition(s)**. **Caution:** Deleting a criticality affects any resource, group or application to which this criticality has been assigned.
   - Edit an existing definition by selecting it in the table, then clicking **Edit business criticality definition**.
   - Create a new business criticality.
   - Start with the default criticalities (Tier 1, Tier 2, Tier 3).
4. Click the **Define criticality +** button to create a new criticality definition, click **Start with presets** (then **Add**), or click the pencil icon next to an existing definition to edit it.
5. Enter or edit a name (required), and description (optional).

   Tip: Use names that reflect your business needs, and that quickly identify the level of importance, for example `Tier 1`, `Level C`, and so on.
6. Enter a criticality value, either by typing a number between 1 and 100, or by using the **+** (plus) and **-** (minus) buttons to select a numerical value.

   Important: The higher the value you specify for the business criticality, the more important it is when assigned to an application (or resource/resource group).

7. Click **Save** (or **Define**). The business criticality definitions are saved and listed on the *Business criticality* page.

Criticality values can now be assigned to applications (or resources/resource groups) from the *Resource management* page or while viewing an application.

Running data administration routines

You can run data management routines from the *Data administration routines* page. You can also view details for any routines that have been run.

Before you begin

You must be an administrator user with **`waiops_manage_topology_routines`** permissions.

About this task

Long-running routines that perform certain data management activities are run by the system as and when required, and you can view details for these when they have run on the Data administration routines page. You can also run some of these manually when required, for example to re-index vertices for searching, or to delete all resources for a specific provider.

You can run the following routines. After defining the details specific to the routine, you specify the number of updates to be batched up, then run the routine to implement its specific function.

- **Deduplicate** - This routine removes duplicate resources from the system.
- **Delete provider resources** - This routine permanently deletes all the resources and resource groups supplied by a specific provider.
- **Encryption key change** - This routine updates passwords in observer jobs after the encryption key has been changed.
- **Rebroadcast** - This routine rebroadcasts vertices from the topology to force a re-indexing of the search database.
- **Resource status delete** - This routine improves system performance by deleting all status for a single resource that has an excessive amount of status assigned.
- **Status clear** - This routine clears obsolete status if there have been synchronization problems with the event management system.

Procedure

1. Log in to the IBM Cloud Pak Automation console.
2. From the main navigation, click **Resource management**.
3. From the **Settings** drop-down, click **Topology configuration**, then click **Configure** on the Data administration routines card.

   The *Data administration routines* page is displayed listing existing routines that have been run. If user-defined display settings are in place, some routines may no longer be displayed. The table footer will indicate if this is the case.

   Tips:

   - To reduce the number of items displayed in the table, use the Filter table text field. The table is filtered as you enter the search text.
   - To reload the items displayed, which can be useful if other users are customizing them, click **Refresh** (top right).

   From here, you can perform the following actions:

   - Sort the table of defined criticality definitions by column header, which are name, type, status (ready or completed), last run, duration, operations (number), or items visited (number).

- View a routine's parameters and parameter values when run last (by clicking the routine name).
- Configure which data administration routines are displayed (by setting the display options).
- Run a routine manually.

4. To define routine display settings, click the **Routine display settings** button (top right). The *Routine display settings* page opens, where you can set the following options:

   Age limit (days) Set the number of days from which a routine will be considered 'old' instead of 'recent'. The default is 30 days.

   Display options Next, select one or more of the following checkboxes to hide the selected routines based on your definitions of 'old' and 'recent':

   - Hide old routines that finished successfully
   - Hide recent routines that finished successfully
   - Hide old routines that were aborted
   - Hide recent routines that were aborted

5. To run a data administration routine manually, click the **Run a routine** button. The *Run a new routine* window opens displaying the available routines that you can run manually.

6. Select the routine to be run, then click **Next**. A window specific to the routine selected is displayed.

7. Define the details for the specific routine to be run.

   Deduplicate This routine removes duplicate resources from the system. Specify the number of updates to be batched up. The default is 1,000 (one-thousand).

   Delete provider resources This routine permanently deletes all the resources and resource groups supplied by a specific provider. Select the provider from a drop-down list of existing providers. Specify the number of updates to be batched up. The default is 1,000 (one-thousand).

   Encryption key change This routine updates the observer job passwords after the encryption key was changed. Enter the old key to be changed. Specify the number of updates to be batched up. The default is 1,000 (one-thousand).

   Rebroadcast This routine rebroadcasts topology data to trigger re-indexing of the search database. Specify whether historic resources should be rebroadcast (**true** or **false**). Enter the broadcast IDs to be rebroadcast. You can add more than one. Choose whether the resource status should be rebroadcast (default is 'true'). Choose whether group and composite edges for resources should to rebroadcast (default is 'true'). Specify the number of updates to be batched up. The default is 1,000 (one-thousand).

   Resource status delete This routine deletes all status for a single resource. Use the Search field to find the resource to be stripped of its status values. Set the minimum age of any status to be deleted. The default is 12 hours. Set a deletion limits (maximum). The default is 1.000 (one-thousand). Specify the number of updates to be batched up. The default is 1,000 (one-thousand).

   Status clear This routine clears all status for the system. Set the minimum age of any status to be deleted. The default is 12 hours. Set a deletion limits (maximum). The default is 1,000 (one-thousand). Specify the number of updates to be batched up. The default is 1,000 (one-thousand).

8. Click **Run**.

When the routine has completed, its details will be displayed on the *Data administration routines* page.

Configuring topology user preferences

Set your personal preferences for how topologies are rendered.

About this task

The settings that can personalize are grouped into the following categories:

- Updates

- Layout
- Miscellaneous

Procedure

You can access the user preferences settings while viewing a topology or from the *Resource manager* page.

1. Log in to the IBM Cloud Pak Automation console.
2. From the main navigation, expand **Operate** and click **Resource management**.
3. Click the **Settings** icon that is next to the search field. Click **User preferences**.
4. Update the setting for any user preference setting that you want to change.
5. Click **Save** when you complete your changes and return to the topology view.

You must reopen the page before any changes to this user preference take effect.

Updates

These settings determine how and how often topologies are automatically updated.

- **Default auto refresh rate (seconds)**

  Set to `30`.

- **Maximum number of resources and relationships to load with auto refresh enabled**

When the set resource limit is reached, auto-refresh is turned off. The maximum value is `2000`. By default, the value is set to `500`.

Tip: If you find that the default value is too high and negatively impacts your topology viewer's performance, reduce this value.

- **Auto render new resources**
  - Enable this option to display new resources at the next scheduled or manual refresh as soon as they are detected. By default, this setting is disabled.
- **Remove deleted topology resources**
  - Enable this option to remove deleted resources at the next scheduled or manual refresh. By default, this setting is disabled.

Layout

These settings determine the default presentation of topologies. You can select the layout type to use for rendering a topology by default. In addition, for some layouts you can select to change the layout orientation for how the layout is displayed.

Tip: A change to a layout type is tracked in the URL (layout orientation is not tracked). You can manually edit your URL to change the layout type display settings.

- **Default layout type**
  - (Default) Use Layout 1
- **Layout 4 orientation**
  - (Default) Top to bottom
- **Layout 6 orientation**
  - (Default) Top to bottom
- **Layout 7 orientation**
  - (Default) Top to bottom
- **Layout 8 orientation**
  - (Default) Top to bottom

Miscellaneous

Other settings, such as displaying messages, screen reader support, and troubleshooting.

- **Information message auto hide timeout (seconds)**

  The number of seconds that information messages are displayed in the IBM Cloud Pak Automation console. The default is 3.

  Tip: If you are using a screen reader, it may be helpful to increase this value to ensure that you do not miss a message.

- **Screen reader support for graphical topology**
  - You can enable the display for more Help text on-screen elements, which can improve the usability of screen readers. By default, this setting is disabled.
- **Client-side debug logging (on supported browsers)**
  - Enhanced client-side ability to log problem diagnosis. If enabled, more debug output is generated, which you can use for defect isolation. By default, this setting is disabled.

  Tip: Use this for specific defect hunting tasks, and then disable it again. If this setting is left enabled, the Topology viewer performance can be reduced.

- **Maintain zoom level on render configuration changes**
  - (Default) Disabled

Configuring topology elements

You can customize a number of IBM Cloud Pak Automation console elements for the Topology viewer, such as resource styles and icons. You can also create custom topology tools which users can access through a topology's context menu.

Tip: The Topology viewer loads all configuration settings into memory when it opens. If you change your topology configurations after opening the Topology viewer, you must reopen it before these changes take effect.

Configuring advanced topology settings

You can define advanced topology settings. Examples include the URLs for trusted sites, the required properties for tooltips or relationship type styles, or the maximum hop numbers a user can choose in the topology viewer.

Configuring topology tools

You can create custom topology tools, which users can then access from within a topology's context menu. Use this feature to access properties of a selected item, such as a resource or relationship, and run some customized functions within the context of that item.

Configuring resource types

You assign existing or new custom icons to particular resource types using the Resource types page. Here you can also apply further resource type style edits, such as adding custom labels to a resource type, and changing its shape, size, border and background.

Configuring relationship types

You can customize the styles and labels for specified relationship types using the Relationship types page. You can also delete existing, or create new relationship type styles.

Configuring custom icons

You can add custom icons for resources that are displayed by using the Custom Icons page.

Configuring rules

Rules help streamline topologies and conserve system resources, such as by merging different observer records of the same resource into a single composite resource, or by excluding specific changes from being recorded against a resource history.

Configuring advanced topology settings

You can define advanced topology settings to control how the topology UI works for all users.

You can change settings, such as the URLs for trusted sites, the required properties for tooltips or relationship type styles, or the maximum hop numbers a user can choose in the topology viewer.

Procedure

1. Log in to the IBM Cloud Pak Automation console.
2. From the main navigation, expand **Operate** and click **Resource management**.
3. Click the **Settings** icon that is next to the search field. Click **Topology configuration**.
4. On the *Advanced setting* card, click **Configure**.
5. From the *Advanced topology settings* page, set or change any of the settings or values to meet your needs.

   The settings are included within three categories:

   - **Topology rendering**
   - **Topology search**
   - **Topology tools**
6. After you have completed your changes, click **Save** to save your changes and return to the *Topology configuration* page.

Available settings

Topology rendering

Within the *Topology rendering* section, you can set or change the following settings:

- **Default hop type for topology viewer**

  Choose the default hop type to be applied when viewing a resource in the Topology Viewer. The default setting is 'element to element'.

- **Maximum number of hops allowed**

  Select a value in the range 2 - 30 from the drop-down list. The default maximum hop count is `4``.

  Important: Ensure that you set a maximum hop count that does not overload your system data capacity when the maximum hop count is selected.

- **Historical changes threshold**

  This setting defines the maximum size of a topology based on the number of resources for which the historical changes indicators are enabled in delta mode.

  Set the maximum topology size by entering a value up to, but not exceeding **50**.

- **Include adjacent relationships on topological hops**

  Select whether to retrieve all relationships between resources within a topology by enabling or disabling the toggle. By default, this setting is enabled and all relationships are retrieved.

- **Properties required for tooltips or resource and relationship styling**

To improve system performance only specific resource and relationship properties are loaded into the topology viewer. You can specify more properties to be retrieved by adding the properties to the table. These required properties can be uses in topology tooltips, custom status tools or to customize UI elements with JavaScript.

To add a property, enter the property name within the text field and click **Add +**.

After you have added a property, you can choose whether the property can be displayed in a tooltip. If you want to exclude the property from tooltips, clear the **Show in tooltip?** checkbox.

- **Resource types to block from topologies**

  Excluded resource types do not display in the Topology Viewer. Excluding resource types can optimize performance when complex topologies are being rendered. You can see the list of excluded types on the *Filter* toolbar of a topology visualization.

  To exclude a resource type from being displayed in the topology viewer, enter the resource type within the text field and click **Add +**.

  Note: You cannot exclude the seed resource of a topology.

  Tip: If an administrator unblocks a resource or relationship type while it is being viewed, the browser in which the topology viewer is displayed must be restarted for the unblocked types to appear.

- **Relationship types to block from topologies**

  Excluded relationship types do not display in the Topology Viewer. Excluding relationship types can optimize performance when complex topologies are being rendered. You can see the list of excluded types on the *Filter* toolbar of a topology visualization.

  To exclude a relationship type from being displayed in the topology viewer, enter the relationship type within the text field and click **Add +**.

- **Properties to hide in the resource details page**

  Excluded properties do not display on the *Resource details* page. In order to reduce visual clutter, you can hide properties that are of no interest to your operator users.

  If a property has been excluded from view by an administrator, it is removed from the tabular view, but remains visible in JSON format.

  To exclude a property from being displayed on the *Resource details* page, enter the property within the text field and click **Add +**.

Topology search

Within the *Topology search* section, you can define the following settings:

- **Add additional properties for filtering**

  By default, users can filter the resources, resource groups, or applications displayed on the *Resource management* page by using the 'entityTypes', 'name' and 'tags' properties. You can add additional properties to be available for filtering.

  Limitation: You cannot define filters by using properties with values that are sets.

  1. Type in a property name, then click **Add+**.
  2. Select the Operators to be available for the property from the Operators drop-down, then click **Add+** again.

  The added properties will be available when operator users define other filter properties on the Resource management page.

- **Include deleted resources in Search results**

  You can select to include or exclude an option in the topology viewer to include deleted resources within topology viewer search results.

By default, the display of the option to include deleted resources is enabled. To disable the display of the option to include deleted resources, switch the toggle to **Disabled**..

Topology tools

Within the *Topology tools* section, you can add trusted websites that can be accessed through HTTP.

If you need to link to an HTTP address instead of HTTPS while writing a custom tool definition, and need to request data from a site that uses HTTP instead of HTTPS, then you can use the UI server as a proxy. The URL to the HTTP proxy takes the actual target HTTP site URL as a parameter. The proxy then sends the request itself, and returns the response back to the browser.

For security reasons, the HTTP proxy can access only the URLs defined by an administrator user as 'trusted sites'.

The **`trustedSites`** property values are a comma-separated list of the trusted sites to which the proxy server can link, and from which it can retrieve information.

Trusted sites operate with a 'starts with' condition.

To add a trusted site, complete the following steps:

1. Within the **Trusted website URL** table, enter the URL of the trusted site.

   For example, if you include the following URLs, the HTTP proxy permits requests for all target URLs that start with included trusted URL.

   - **`www.ibm.com`**
   - **`data-server.intranet.com:8080/network/statistics`**

   For instance, you can use the proxy for URLs, such as **`www.ibm.com/cloud`** or **`data-server.intranet.com:8080/network/statistics/1893713/info`**.

   You cannot use the proxy for URls such as **`www.ibm.co.uk`** or **`data-server.intranet.com:8080`**

2. Click **Add +**.

Configuring custom tools

You can create custom topology tools, which you and other users can then access and use from the context menu of a topology. Use this feature to access the properties of a selected item, such as a resource or relationship, and run some customized functions within the context of that item.

Before you begin

Ensure that you are proficient in JavaScript. You must use JavaScript to define custom tools.

About this task

Custom tools can be accessed through the right-click context menu in the UI. All users with permission to access topologies can access the tools, but only administrators with permission to manage topology tools can customize them.

Note: Some topology tools might fail due to browser restrictions, such as when the tool needs to load external content. Security measures for Firefox and Chrome, which prevent click-jacking and similar risks, can also prevent the loading of third-party content over HTTPS into an iframe. An iframe mechanism can be used by custom topology tools to display contextual information, and as such can be affected. As a workaround for Firefox browsers, users can select an option to load the content in a separate window.

Procedure

1. Log in to the IBM Cloud Pak Automation console.

2. From the main navigation, click **Resource management**, then access a topology.
3. In the navigation toolbar for a topology, click the **Settings** icon. From the drop-down menu, click **Topology configuration**. The *Topology configuration* page is displayed. The page includes cards for each category of topology elements and settings that can be configured or customized.
4. On the *Topology tools* card, click **Configure**. The *Topology tools* page is displayed, listing all of your defined tools in table format in sortable columns. These are the tool name, menu label, last update, the items the tool applies to, whether special conditions exist, and its description.

   From here you can create a new topology tool, or select existing tools in the table and edit or delete them.

   Tip: To reduce the number of items displayed in the table, use the Filter table text field. The table is filtered as you enter the search text. You can also click **Refresh** (upper right) to reload the items displayed, which can be useful if other users are customizing them.
5. Click **New** to create a new tool. The New topology tool page is displayed consisting of the expandable Details, Implementation, and Conditions sections.
6. In the **Details** section, enter the general details for your new tool:

   - **Name**

     Required. Enter a unique internal name for your tool. This name is used as an internal reference for identify individual tools.

   - **Menu label**

     Required. The menu label is the text that is shown in the context menu. This can be the same name as used by other tools, which is why the unique name is required.

   - **Description**

     Optional. A description to help administrative users record the tool's purpose. This text does not display in the context menu.

   - **Menu priority**

     Optional. The menu priority slider defines where in the context menu the tool is displayed. Available values are one to ten.

     For example, tools with a priority of two will be displayed higher in the menu than tools that have a priority of four.
7. In the **Implementation** section, define the JavaScript implementation for your tool.

   If needed, you can click **Example** to view example JavaScript to show you how to format your code. The code for your tool runs when the tool is accessed by an operator, such as when a user from the context menu of a topology.

   To help you create tools, you have access to custom helper functions
8. In the **Conditions** section, select any conditions for using the tool.

   Select the resources, relationships, and status for which the tool applies. A user must select one of these items for the tool to display in the context menu.

   Use the **Applicable item type for tool definition** field to select between the types of topology items for which the tool is applicable:

   - **Resource**

     Select one or more resource types from the list displayed.

   - **Relationship**

     Select one or more relationship types from the list displayed.

- **Resource and Relationship**

  Select one or more resource or relationship types from the combined list.

- **Status**

  Select from the following possible states for which the tool will be available:

  - **Clear**
  - **Indeterminate**
  - **Information**
  - **Warning**
  - **Minor**
  - **Major**
  - **Critical**

Depending on your selection, a number of checkboxes are displayed, which you can use to configure which resources, relationships or states are included. If you select **Resource**, **Relationship**, or **Resource and Relationship** (but **not** Status), a JavaScript editor is displayed, letting you define additional conditions for the display of the topology tools.

The following relationship example creates a tool that shows only relationships that have source resources with the name 'kafkaService', or relationships that have target resources with 'entityType' properties that also include 'container':

```
if (asmSourceProperties.name === "kafkaService" ||
asmTargetProperties.entityType.includes("container")) { return true; } else { return
false;}
```

The following relationships and resources example is for relationships and resources that have a property that is called 'kubernetes_namespace':

```
if (asmProperties.kubernetes_namespace) { return true; } else { return false; }
```

For each category, you can also select **All types** if you want the tool to be displayed for all types under the selected category. The tool also displays for any specific types under the selected category that is not listed.

Remember: When you are creating status tools, the properties that you use in your JavaScript reference the properties for the status item. The properties that you use in your resource or relationship tools reference the properties for the resources or relationships.

9. Click **Save** to save your configuration and create the tool.

   Once you save your changes, you must close the Topology Viewer, and then reopen the UI to make the new tool available. Tools are also available on the context menu of a topology only when the configured conditions are met.

Custom helper functions

To help you create custom topology tools, you can use the following custom JavaScript helper functions:

asmProperties

Your tool implementation can access the properties of the relevant resource, relationship, or status through the **`asmProperties`** JavaScript object, which contains all the properties.

You can access the properties by using standard JavaScript, but you must protect against the property value not being present. For example, if you intend to use the property 'latitude', you must verify that it is present before using it. To do so, you can use the following check command:

```
asmProperties.hasOwnProperty('latitude')
```

If the property is present, the Boolean value true will be returned.

Status-related tool properties

For **Status**-related tool properties, you use JavaScript that is similar to the JavaScript that you use when creating resource or relationship tools. However, the properties that you must use in your status tool implementation, such as **`asmProperties`**,

must reference the properties for the status item. For **Resource** or **Relationship** related tools, the implementation needs to instead reference the properties for the resources or relationships.

For example, if you use `asmProperties.location` in a status tool implementation, a corresponding `location` property must be in the status record. When creating status tools, the `asmProperties` object has a property that takes the form of an array that is called `resources`, which represents the resources in the topology with which this status is associated. Each item in the `resources` array is an object with properties that represent the properties of that resource. If a status is associated with two resources, the `uniqueId` property of the first of those two resources can be referenced in the JavaScript by using `asmProperties.resources[0].uniqueId`.

In addition, you can access the properties of the resource for which you are intending to use a status tool by using the `asmSourceProperties` object when scripting the status tool.

asmSourceProperties

You can access information about the source properties of any relationship or status that the custom tool is acting upon through the `asmSourceProperties` JavaScript object.

The following example shows the use of the helper function in a custom relationship stroke definition:

```
if (asmSourceProperties.myProp === 'high') {
    return 'blue';
} else {
    return 'black';
}
```

Note: The arrows on the topology indicate a relationship point from the source to the target.

asmTargetProperties

You can access information about the target properties of relationships that the custom tool is acting upon trough the `asmTargetProperties` JavaScript object.

asmFunctions

You can use a number of other helper functions, which are accessed from the asmFunctions object, which includes the following functions:

showConfirmationPopup(title, message, onOk)

Creates a confirmation dialog to allow the tool to confirm an action. This function takes a title and message, which is displayed on the dialog, and a function definition, which is run when the user clicks the **OK** button on the dialog.

showToasterMessage(status, message)

Opens a window toaster message with the appropriate status coloring and message.

showPopup(title, text)

Shows a dialog with a title and text body, which can be generated based on the properties of the resource or relationship.

Tip: With the `asmFunctions.showPopup` helper function, you can use markdown to create more sophisticated HTML pop-up windows. For more information about markdown syntax, consult a reputable markdown reference site.

showIframe(url)

Displays a dialog that fills most of the page, which wraps an iframe, showing the page of the URL. Use this function to embed more pages.

Restriction: You must add any websites that are referenced by the `url` parameter to a list of your trusted sites. For more information about adding sites to this list, see Defining advanced topology settings.

getResourceStatus(<resource_id>, <callback_function>, [<time_stamp>])

Use this function to request status information from a tool definition for a resource by using its `_id` parameter.

- **`resource_id`**

  Required. This value can be obtained from a resource through **`asmProperties._id`** and from a relationship by using **`asmSourceProperties._id`** or **`asmTargetProperties._id`**.

- **`callback_function`**

  Required. This function is called when the status data is collected from the topology service, with a single argument that contains an array of status objects.

- **`time_stamp`**

  Optional. A Unix millisecond timestamp to get the status from a set point in time.

The following example shows the status information of a source resource from a relationship context to the browser console log:

```
let printStatusCallback = function(statuses) {
    statuses.forEach(function(status) {
        console.log('status:', status.status,
                    'state:', status.state,
                    'severity:', status.severity,
                    'time:', new Date(status.time));
    })
}
asmFunctions.getResourceStatus(asmSourceProperties._id, printStatusCallback);
```

sendHttpRequest(url, options)

Use this function to send an HTTP or HTTPS request to a remote web server by using the IBM Cloud Pak® for AIOps backend server rather than the browser to avoid browser domain-blocking.

- **`url`**

  Required. The full URL of the remote site to be accessed, for example:

  `https://data-svr-01.uk.com/inv?id=1892&offset=0`

  Restriction: You must add any websites that are referenced by the **`url`** parameter to a list of trusted sites. For more information about adding sites to this list, see Defining advanced topology settings.

- **`options`**

  Optional parameters that you can include:

  - **`method`**

    The HTTP method to use. You can use the following methods:

    - GET (default)
    - POST
    - PUT
    - DELETE

  - **`headers`**

    An object that defines any required special request headers.

  - **`body`**

    A string that contains the body data for the request. Use POST and PUT requests only.

  - **`autoTrust`**

    A flag to indicate whether the remote web server can be automatically trusted. This flag is required when the website uses a self-signed SSL certificate with no CA, or uses a CA that is unknown to IBM Cloud Pak for AIOps. The value for the flag can be **`true`** or **`false`**. The default value is **`false`**.

  - **`onSuccess`**

A callback function to run if the HTTP request succeeds. This function is passed the following three parameters:

- Response text
- HTTP status code
- Response header

- **onError**

  A callback function to run if the HTTP request fails. This function is passed the following three parameters:

  - Response text
  - HTTP status code
  - Response header

The following sample JavaScript, shows the use of the **options** parameter:

```
{
    method: 'GET',
    headers: {
        Content-Type: 'application/json',
        X-Locale: 'en'
    }
    body: '{ "itemName": "myData1" }',
    autoTrust: true,
    onSuccess: _onSuccessCallback,
    onError: _onErrorCallback
}
```

Example tools

Use the following examples to help you create the JavaScript implementation for your own custom topology tools. This implementation needs to define the action that occurs when a user selects the tool from the context menu.

Example: Send HTTP GET request for external REST data

The following example uses the **asmFunctions.sendHttpRequest(url, options)** helper function to send an HTTP GET request for external REST data, which is then displayed in a dialog on the Topology viewer.

Reminder: You must add any websites that are referenced by the **url** parameter to a list of trusted sites. For more information about adding sites to this list, see Defining advanced topology settings.

```
// Get the unique id from the resource and use it to send a REST request
var extId = asmProperties.uniqueId;
if (!extId || extId === '') {
    asmFunctions.showToasterMessage('warning', 'The resource does not have a unique
identifier.');
} else {
    var options = {
        method: 'GET',
        onSuccess: _onSuccess,
        onError: _onError
    };
    asmFunctions.sendHttpRequest('https://api.data.uk/resource/' + extId, options);
}

function _onSuccess(response, status, headers) {
    // Start to build text for the info dialog
    var output = 'Here are details for ' + asmProperties.name + ':\n\n';

    var data;
    try {
        data = JSON.parse(response);
    } catch (e) {
        data = {};
    }
    var props = data.additionalProperties;
    if (!props || props .length === 0) {
        asmFunctions.showToasterMessage('information', 'No information exists for this
resource.');
    } else {
```

```
        for (var idx in props) {
            var prop = props[idx].key;
            var val = props[idx].value;
            if (!prop.endsWith('Id')) {
                output += props[idx].key + ' = ' + props[idx].value + '\n';
            }
        }

        // Show the output in an ASM popup dialog
        asmFunctions.showPopup('External Details', output);
    }
}

function _onError(response, status, headers) {
    if (!response || response === '') {
        asmFunctions.showToasterMessage('critical', 'HTTP request failed with status code ' +
status);
    } else {
        asmFunctions.showToasterMessage('critical', 'HTTP request failed: ' + response);
    }
}
```

Example: Splunk integration

When you are configuring a custom tool, you can create a custom tool integration with Splunk. As an example, you can create an integration that allows you to search a Splunk repository through the context menu of a selected resource in the Topology viewer. For more information, see Example: Splunk integration.

Splunk integration example

With IBM Cloud Pak® for AIOps, you can add custom tools to the topology viewer, such as a custom tool integration with Splunk. The following example creates a custom tool for searching a Splunk repository. This tool can be accessed from the right-click context menu on the topology viewer when you are viewing a resource topology.

Note: You must install Splunk Enterprise and Splunk Forwarder before you can define a custom tool with the following example. The following example was tested with Splunk 8.1.0. For more information, see Splunk.

About this example

The following example code creates a custom tool that can be accessed from the right-click context menu. With this tool, you can use the properties of the resource that is selected in a topology when you click to open the tool to search a specific Splunk repository. This specific tool searches Splunk for logs that match the name of an IBM Cloud Pak for AIOps resource. This search can help you determine whether Splunk has registered any log information about the selected resource.

If you want to customize the following example code, you can change the values for the following properties:

- **`splunkUrl`**

  The base URL of your Splunk instance

- **`splunkRepository`**

  The Splunk repository you want to search

- **`searchValue`**

  The dynamically created **`searchTerm`** used in Splunk

In addition, you can use any IBM Cloud Pak for AIOps properties that are associated with the resource to create your tool. You can also define the tool response for when a search is unsuccessful.

Example

```
// Set the Splunk instance url
var splunkUrl = 'host:port';
// This will store the value that you want to search in Splunk
var searchValue = '';
// This will search results for defined period
//    0 = all time
// -15m = last 15 mins
// -24h = last 24 hours
//  -7d = last 7 days
var searchEarliest = "0";
// This flag is set if you find a valid search value
var foundValidSearchValue = false;
if (asmProperties && asmProperties.name) {
    searchValue = "\"" + asmProperties.name + "\"";
    foundValidSearchValue = true;
} else {
    var status = 'critical';
    var message = 'Unable to find name property for Splunk search request';
    asmFunctions.showToasterMessage(status, message);
}
// DO NOT MODIFY THE LINES BELOW
if (foundValidSearchValue) {
        var searchLatest = '';
        if(searchEarliest!="0") {
                searchLatest="now"
        }
    var encodedSearchValue = encodeURI(searchValue);
    var searchQuery = '/en-US/app/search/search?
q=search%20index%3D*%20'+encodedSearchValue+"&earliest="+searchEarliest+"&latest="+searchLates
t;
        window.open(splunkUrl+searchQuery);
}
```

Configuring resource types

You can assign existing or new custom icons to particular resource types by using the Topology viewer *Resource types* page in the UI console. You can also apply further resource type style edits, such as adding custom labels to a resource type, and changing the resource type icon shape, size, border, and background.

You can define a custom icon as part of editing a resource type to have the icon immediately assigned to the selected resource type. However, if you want to create a number of icons without assigning them to specific resource types, or you simply want to edit or delete icons, use the *Custom icons* page. For more information, see Defining custom icons.

Procedure

1. Log in to the IBM Cloud Pak Automation console.
2. From the main navigation, expand **Operate** and click **Topology viewer**.
3. From the topology navigation toolbar, expand **Settings**, and click **Topology configuration**.
4. On the *Resource types* card, click **Configure**.

   The Resource Types page is displayed, which lists all existing resource types in table format in sortable columns. The table also displays the icons for the resource types and their names, categories (if defined), whether they are system icons or custom icons, and whether they have custom labels, styles or shapes. From here, you edit existing resource types, and then associate existing icons with them, and apply resource styling.

   Tips: To reduce the number of items displayed in the table, use the Filter table text field. The table is filtered as you enter the search text. You can click **Refresh** (top right) to reload the items displayed, which can be useful if other users are customizing them. You can select more than one resource type and reset them to their default values, or perform a batch edit.

5. To edit resource types, select them in the table and click **Edit resource type** (or **Batch edit resource types**). The *Configure relationship type* page is displayed.

   The *Edit Resource Type* page is displayed. You can toggle between the **Identification** and the **Styling** tabs. Use the options on these tabs to define the resource type.

   - Use the **Identification** tab to define the selected resource type's icon, label, and shape.
   - Use the **Styling** tab to define the selected resource type's border color, border pattern, background color, and resource display size.

   Restriction: You cannot change the name of an existing resource type.

   Tip: To quickly revert resource stype styles to their default, select them in the table, then click **Reset to defaults**.

6. Click **Save** when you are finished with your edits to save your changes and return to the *Resource types* page. The table displays your changes for the resource type and the associated icon.

Identification

On the **Identification** tab, define the selected resource type icon, label, and shape:

- Choose an icon to associate with the resource type by using one of the following methods:
  - **Icon** drop-down list

    If you open the **Icon** drop-down list, all icons are listed by name in alphabetical order.
  - **View all icons**

    If you click **View all icons**, all icons are displayed in alphabetical order. Click an icon to associate it with the resource type.
  - **Quick assign**

    If an icon exists with the same name as the resource type, click the **Quick assign** button to select it without having to sort through all existing icons. This function is useful if you have added a custom icon, and are now assigning it to a resource type with the same name.
  - **Define new custom icon**

    From here you can define a custom icon, which is automatically associated with the resource type when done.

    - Click the **Define new custom icon** button to display the Custom Icons page.
    - Click **New** to create a new icon. Alternatively, click the *edit* symbol to edit an existing icon. After you have selected or created an icon, the *Configure resource type* page is displayed. Use the following information to define the icon:

      Icon properties

      Each custom icon must have a name that uniquely identifies the icon when assigning it to a type.

      Remember: You cannot change the name of an existing icon. If you want an icon to have a different name, create a new icon, then delete the old one.

      Icons are defined inside an SVG editor, which runs an XML validation check. Each icon definition must be valid SVG XML with a viewBox, which is important to ensure scaling of the image.

      The SVG definition must include inline styling of the image, such as stroke color and fill color. Use of style classes is not advised, as it can cause visualization issues on some browsers. If style classes must be used, naming must be unique for each svg image to prevent class definitions from being overwritten.

      Optionally, each icon can be assigned to a category, so you can group icons of the same type or function together when displaying them in a list.

Remember: You can also create custom icons from the Custom Icons page accessed through DASH, which is described in the Defining custom icons topic.

- Define the label for the resource type by editing the following fields:
  - Label

    By default the name of a property (**`asmProperties.name`**) is used as the label that is displayed in the topology. You can replace this by typing in a custom label for the resource type.
  - Label Maximum Length

    You can override the default label length of 15 characters to avoid truncating the displayed label.

    Warning: Avoid labels that are too long, as long labels can overlap in the topology. The maximum suggested label length is 20 characters.
- Choose a shape for the resource type from the **Resource Shape** drop-down list.

  The default shape for a resource in the Topology Viewer is a circle, or a square for a host server. You can change the shape of the resource type to one of the following shapes:
  - Circle
  - Square
  - Hexagon
  - Vertical hexagon

Styling

On the **Styling** tab, you can define the selected resource type border color, border pattern, background color, and resource display size:

- For the border color, enter a hex definition. The default border color is **`#171717`**.
- For the border pattern, the default pattern is `''`.
- For the resource background color, the default is **`#F3F3F3`** (light gray).
- For the size, the default is medium.

  You can further refine the size function by specifying how certain resource properties affect the size of the resource type displayed. For example, resource types can appear larger depending on the number of connections they have with other resources.

Configuring relationship types

You can customize the styles and labels for specified relationship types using the Relationship Types page. You can also delete existing, or create new relationship type styles.

You customize styles in two steps. First, you identify the relationship type to be customized, or create a relationship type. Then, you can customize the style elements for the relationship type. To customize the style, you can set the relationship line color, thickness, style, and label.

Any resource or relationship properties that are used in custom relationship styles must be added as required properties to the *Advanced topology settings*. If the properties do not exist in the advanced topology settings, you are prompted to add the properties when you save a new relationship style. For more information, see Advanced topology settings.

Procedure

1. Log in to the IBM Cloud Pak Automation console.
2. From the main navigation, expand **Operate** and click **Topology viewer**.

3. From the topology navigation toolbar, expand **Settings**, and click **Topology configuration**.
4. On the *Relationship types* card, click **Configure**.

   The *Relationship types page* is displayed and lists all of your existing customized relationship types in a table with sortable columns. The table also displays the last updated timestamp, and indicates whether a relationship type has a custom label, custom color, or custom width defined. From here, you can choose to delete or edit configurations. You can also create a relationship type configuration.

   Tips: To reduce the number of items displayed in the table, use the Filter table text field. The table is filtered as you enter the search text. You can click **Refresh** (top right) to reload the items displayed, which can be useful if other users are customizing them. You can select more than one relationship type and reset them to their default values, or perform a batch edit.
5. Select the action that you want to complete:

   - To delete a relationship type, select them in the table and click **Delete relationship settings**.
   - To revert to default settings, select an item in the table and click **Reset to defaults**.
   - To edit existing relationship types, select them in the table and click **Edit relationship type** (or **Batch edit relationship types**). The *Configure relationship type* page is displayed.
   - To create a new relationship type, click **New**. The *Configure relationship type* page is displayed.
6. Select the **Identification** tab on the *Configure relationship type* page.

   - Relationship Type

     Choose the relationship type that you want to configure from the dropdown list.

     Note: Only relationship types that exist in your topology database are listed.

   - Label Function

     Example of a JavaScript function to define a label for the relationship using the resource properties:

     ```
     return asmProperties.labelText;
     ```
7. Select the **Style** tab on the *Configure relationship type* page.

   Warning: If you use the advanced JavaScript format to define colors, do not define a numeric value instead of an actual color, as this will prevent the topology rendering from being completed.

   - Line Color Function

     Example JavaScript function to define link color:

     ```
     return 'blue';
     ```

     Default function:

     ```
     return '#171717';
     ```

   - Line Width Function

     Example JavaScript function to define line width for the relationship using the source resource properties:

     ```
     return asmSourceProperties.myProp > 10 ? '9px' : '1.5px';
     ```

     Default function:

     ```
     return '0.5px';
     ```

   - Line Pattern Function

     Example of a JavaScript function to define the line pattern:

     ```
     return '3 4 5';
     ```

     Default function:

     ```
     return null;
     ```

8. Click **Save**.

   If any resource or link properties are not defined in the *Advanced topology settings*, you are prompted to save the properties now.

The relationship style and label for the specified relationship type is customized.

Style restrictions

- The line color (strokeWidth) property can have a maximum value of 9px. Any higher value can cause styling problems.
- The label (linkLabel) property must return a valid string to be displayed alongside the relationship.

For detailed SVG style definition information, see:

- Line color (stroke)
- Line width (stroke-width)
- Line pattern (stroke-dasharray)

Accessing properties for styling functions

When you are defining styles, you can access dynamic properties of the relationship or connected resources.

- To access the properties of resources, use the **`asmProperties`** JavaScript object.
- To access the properties of relationships, use the **`asmSourceProperties`** or **`asmTargetProperties`** JavaScript objects.

  Remember: The arrows indicating a relationship point from the source to the target.
- To access the highest open severity status from the source or target resource, use the **`asmSourceProperties._hasStatus`** or **`asmTargetProperties._hasStatus`** JavaScript objects. The following example uses the **`_hasStatus`** parameter to modify the relationship label:

```
if (asmSourceProperties._hasStatus || asmTargetProperties._hasStatus) {
    // object of all ASM's status severities
    let severityRank = {
        clear: 0,
        indeterminate: 1,
        information: 2,
        warning: 3,
        minor: 4,
        major: 5,
        critical: 6
    };
    let sourceSeverityRank = severityRank[asmSourceProperties._hasStatus] || 0;
    let targetSeverityRank = severityRank[asmTargetProperties._hasStatus] || 0;
    let labelString = asmProperties._edgeType;
    // Show highest Status on relationship label
    if(sourceSeverityRank > targetSeverityRank) {
        labelString += ': Status = ' + asmSourceProperties._hasStatus;
    } else {
        labelString += ': Status = ' + asmTargetProperties._hasStatus;
    }
    return labelString;
} else {
    // No status for source or target resources use plain label
    return asmProperties._edgeType;
}
```

Configuring custom icons

You can add custom icons for resources that are displayed in the IBM Cloud Pak Automation console using the *Custom Icons* page.

Before you begin

To access the *Custom Icons* page, you must have the required permissions.

About this task

Tip: Instead of using the following procedure, you can first select a specific resource type from the Resource Types page, and then define a custom icon for that specific resource type only. This is described in Editing resource type styles. However, if you intend to create a number of icons without assigning them, or simply want to edit or delete icons, use the following procedure.

Icon properties

- Each custom icon must have a name that uniquely identifies the icon when assigning it to a type. You cannot change the name of an existing icon. If you want an icon to have a different name, create another icon, then delete the old icon.
- Icons are defined inside an SVG editor, which runs an XML validation check. Each icon definition must be valid SVG XML with a `viewBox`, which is important to ensure scaling of the image.

  The SVG definition must include inline styling of the image, such as stroke color and fill color. Use of style classes is not advised, as it can cause visualization issues on some browsers. If style classes must be used, naming must be unique for each svg image to prevent class definitions from being overwritten.
- Optionally, each icon can be assigned to a category, which allows you to group icons of the same type or function together when displaying them in a list.

Procedure

1. Log in to the IBM Cloud Pak Automation console.
2. From the main navigation, expand **Operate** and click **Topology viewer**.
3. From the topology navigation toolbar, expand **Settings**, and click **Topology configuration**.
4. On the *Custom icons* card, click **Configure**. The Custom icons page lists the custom icons in table format by name, icon, category, last update, and size, in sortable columns.

   Tip: To reduce the number of items displayed in the table, use the Filter table text field. The table is filtered as you enter the search text. You can also click **Refresh** (top right) to reload the items displayed, which can be useful if other users are customizing them.
5. Click **New** to create an icon. Alternatively, select icons in the table to edit or delete them.
6. On the *New custom icon* page, enter the details for your custom icon:
   - Enter a unique name. Each icon must have a name, which uniquely identifies the icon when assigning it to a type.

     Remember: You cannot change the name of an existing icon. If you want an icon to have a different name, create an icon, then delete the old icon.
   - Optionally, enter a category name. Each icon can be assigned to a category. You can use categories to group icons of the same type or function together.
   - Enter the SVG XML to define the icon. Use the editor to enter valid XML.

     Each icon definition must be valid SVG XML with a viewBox, which is important to ensure scaling of the image. The SVG editor rejects any invalid XML entered.

     The XML editor includes a Preview area where the results of your edits display. The following conditions apply:
     - Height or width should not be specified.
     - Inline styles, such as stroke color and fill color, should be used and not style classes. Naming must be unique for each SVG image to prevent class definitions from being overwritten.

- SVG namespace is required: **`xmlns="http://www.w3.org/2000/svg"`**

Example: Use the following definition for the **`disk`** icon as guidance:

```
<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 64 64">
   <ellipse style="fill-opacity:0;stroke:currentColor;stroke-width:12.12270069;"
id="path4139" cx="33.627117" cy="32.949142" rx="16.803904" ry="17.210684"/>
   <circle cx="33.827423" cy="33.055576" r="3.3037829"/>
</svg>
```

7. Click **Save**.

What to do next

You can assign your custom icons to particular resource types using the *Resource types* page. On that page, you can also apply further style edits to the resource types.

Configuring rules

Rules help streamline topologies and conserve system resources, such as by merging different observer records of the same resource into a single composite resource, or by excluding specific changes from being recorded against a resource history.

Before you begin

Ensure that you have the details to hand about the resources for which you intend to develop rules. For example, to create merge rules, you must know which resources exist as duplicate records before you can merge these into composites.

Rule types

You can use a number of different types of rules for different purposes. The rule type (**`ruleType`**) can be one the following rules.

mergeRule

A merge rule populates the tokens of resources. This is matched by the rule to prevent duplicate records of the same resource from being displayed in the topology. Merge rules are applied to a resource in an observer job **before** it is sent to the topology service.

Different observers can record and then display the same resource as two (or more) resources. To prevent this, you can create a merge rule to ensure that the separate records of the same resource share values in their tokens set, which then triggers the Merge service to create a single composite resource vertex. Composite (or merged) resources are displayed in the Topology viewer as a single resource, which includes the properties of all merged resources.

Example: Default merge rule defined for the Docker Observer

The rule name in this example is **dockerId**. This rule applies only to resource of the type **Docker Observer**. It applies to all entity types and to all providers, and it copies the **dockerId** property into the merge tokens for all resources that have this property set.

Field	Value
Rule name	dockerId
Status	Enabled
Tokens	dockerId
Resource type	*no value specified*
Observers	Docker Observer
Providers	*no value specified*

Example: ALM Observer merge rule

The name of this example rule is **almExternalId**. This rule applies to all resource types and providers, but is applied only when the **ALM Observer** is run. It copies the **externalId** property into the merge tokens for all resources that have this property set.

Field	Action
Rule name	Enter **almExternalId**
Status	Toggle **Enabled**
Tokens	Enter **externalId**
Resource type	Leave empty to apply to all types
Observers	Select **ALM Observer**
Providers	Leave empty to apply to all providers

tagsRule

The tags rule allows the value of another field to be copied to the tags field so that it becomes searchable in the UI. It populates the tags of resources matched by the rule.

Any field that is not indexed and cannot normally be searched becomes searchable when copied to the tags property. For a list of indexed properties, see Properties.

Example: Simple tags rule

The name of this example rule is **tagRuleCustomProp**. This rule applies to all resource types, observers and providers.

Field	Action
Rule name	Enter **tagRuleCustomProp**
Status	Toggle to **Enabled**
Tokens	Enter a tag name
Resource type	Leave empty to apply to all types
Observers	Leave empty to apply to all observers
Providers	Leave empty to apply to all providers

Example: Docker Observer 'container' tags rule

The following example creates tags rules for the Docker Observer and will add **`com.docker.compose.service`** and **`com.docker.compose.project`** properties to the tags list for entities of the type container:

Field	Action
Rule name	tag-docker-container-service-project
Status	Toggle to **Enabled**
Tokens	Type **com.docker.compose.service** and **com.docker.compose.project**
Resource type	Select **Container**
Observers	Select **Docker Observer**
Providers	Leave empty to apply to all providers

Example: ITNM Observer custom tags

You can create custom tags on IBM Tivoli Network Manager, which the ITNM Observer can integrate into the tags fields. For information on creating custom tags in Network Manager, see the Adding name-value pairs to entities using the File finder topic in the IBM Tivoli Network Manager IP Edition documentation.

Having added a custom tag in Network Manager, you can extract the value into your tags field. The following example uses a regular expression to extract the value of the **`Country`** custom tag:

```
customTags=~/[\\{]Country=(.+?)[,\\}]/
```

matchTokensRule

A match token rule allows the value of another field to be copied to the matchTokens field so that it can be used to match events to resources. It populates the `matchTokens` of resources matched by the rule.

Example: Simple match token rule

Field	Action
Rule name	Enter **matchRule**
Status	Toggle to **Enabled**
Tokens	Enter a match token name
Resource type	Leave empty to apply to all types
Observers	Leave empty to apply to all observers
Providers	Leave empty to apply to all providers

historyRule

A history rule identifies properties that change every observation, but that don't indicate an important change, for example a host's `sysUpTime` property. Use a history rule to exclude properties from being retained in history, saving resources by not maintaining detailed historical records of changes to these properties.

Note: By default, a mergeRule and an eventsFilterRule are included for the **Jenkins Observer**.

Procedure

1. Log in to the IBM Cloud Pak Automation console.
2. From the main navigation, expand **Operate** and click **Topology viewer**.
3. From the topology navigation toolbar, expand **Settings**, click **Topology configuration**, and then open the *Rules administration* page.
4. Select one of the tabs (**Merge**, **Match tokens**, **Tags**, **History**, or **Business criticality**).

   Tips: To reduce the number of items displayed in the table, use the Filter table text field. The table is filtered as you enter the search text. To reload the items displayed, which can be useful if other users are customizing them, click **Refresh** (top right).

   From here, you can do the following:

   - To delete rules, select them in the table and click **Delete rule configurations**.
   - To edit an existing rule, select it in the table and click **Edit rule configuration**. The *Edit rule* page is displayed.
   - To create a new rule, click **New**.
5. Define the rule properties:

   - **Rule name**

     Enter the name of the rule, which must be unique within the context of the tenant.

   - **Status**

     Toggle the rule status to either **Enabled** or **Disabled**. When observer jobs are run, they apply only the rules that are in an enabled state.

   - **Tokens**

     Enter the tokens to specify how the rule will be applied.

     For **merge rules**, for example, the tokens contain the list of resource parameter names that become merge tokens for the resources to which the rule is applied. Merge tokens can be constructed using variable substitutions, which allows you to combine more than one property value in a token, and also combine them with literal strings.

     Important: These tokens are the shared elements that the duplicate records to be merged have in common.

- **Resource type**

  To specify the resource entity types for which this rule is valid, either enter one or more values, or select the type from the drop-down list. If you do not specify any entity types, the rule will be applied to all types.

- **Observers**

  Select the observer to which this rule applies from the drop-down list. If you do not specify an observer, the rule will be applied to all observers.

- **Providers**

  To specify the provider to which this rule applies, either enter a value or select the provider from the drop-down list. If you do not specify any providers, the rule will be applied to all providers.

  You can use the mutually exclusive **`excludeTokens`** and **`includeTokens`** properties to filter providers.

- **Exclude tokens**

  These properties discard any tokens that match the entered value.

- **Include tokens**

  These properties apply to a token only when the value matches the entered value.

  Remember: Rules are applied before each observer job is sent to the topology service.

Managing incidents and alerts

IBM Cloud Pak® for AIOps enables you to identify health issues across your application, services, and network infrastructure on a single management console. A **Incident list** displays each incident's priority, status, and description. Drill down into an incident to see the probable cause alerts, alert topology, and associated automations.

The Alert Viewer brings together alert details, alert severity, and alert filtering. Some alerts in the console are presented in alert groups, based on analytics. Using the alert list you can explore why these alerts were grouped together, and this can further help with alert resolution.

For more information about incidents and alerts, see the following topics:

- About events, alerts and incidents
- Incidents
- Alerts

About events, alerts and incidents

IBM Cloud Pak® for AIOps represents the state of managed entities using the following three concepts:

- **Events**

  An event is a record containing structured data summarizing key attributes of an occurrence on a managed entity, which might be a network resource, some part of that resource, or other key element associated with your network, services, or applications.

  An event might or might not indicate something anomalous and is a point-in-time, immutable statement about the entity in question.

- **Alerts**

  Alerts represent an ongoing anomalous condition against a single managed entity. Unlike events, alerts might evolve over time as the condition changes. Alerts have a start and an end time. The creation and evolution of alerts are

informed by events.

- **Incidents**

  Incidents represent the context around an issue which is currently severely impacting operations. This includes all alerts that are related to the issue and information about how the affected resources are related. The creation and evolution of incidents are informed by alerts.

Alert creation

Alerts are created when one or more events indicate an anomalous condition. This is determined based on the `type.eventType` field of an event being set to `problem`. When such an event is received by the system, if no alert already exists for the condition, a new alert is created.

Where multiple events occur, indicative of the same condition on the same resource, the events update an existing open alert where one exists rather than creating a new one. This is determined based on events having the same values for the `resource` and `type` fields as the event that originally created the alert.

If a matching event occurs with a `type.eventType` set to `resolution`, the alert changes to the `clear` state. It stays in this state for 2 minutes. During this time, if another matching `problem` event occurs, the alert re-opens. If however, no more matching events occur, the alert enters the `closed` state. Once an alert is in the `closed` state, it can no longer be re-opened and any subsequent events will create a new alert. **Note**: Alerts that are closed are purged from the active store after 2 minutes.

Alert correlation

IBM Cloud Pak® for AIOps automatically correlates your alerts to determine what alerts are likely to share a common cause. This is determined based on a combination of the following mechanisms:

- **Scope-based correlation** - Any alerts which have the same value for the `resource` field are correlated.
- **Temporal correlation** - The system continually analyzes past alerts to determine which alerts tend to frequently co-occur. When these alerts occur together again, they are correlated.
- **Topological correlation** - Any alerts which refer to resources within the same Resource group are correlated.

For all of these mechanisms, the correlation only occurs where the last related alert was created within the past **15 minutes**. If an alert is created more than 15 minutes after the last alert, they are not correlated.

Incident creation

Incidents are created when you enable one of the two preset default incident creation policies in the Policies UI, or create your own custom incident creation policy. The preset incident creation policies are disabled by default.

Note: If you are upgrading a deployment of Cloud Pak for AIOps from version 3.2.1 or version 3.2.0, the "Default incident creation policy for high severity alerts" remains enabled by default. This is to retain the same behavior as before the upgrade.

IBM Cloud Pak® for AIOps will determine which of your alerts were caused by the same problem. If a relevant incident already exists for a given problem, a new incident will not be created. An existing incident is deemed relevant if it was created due to any other alert which shares a common cause with the alert in question. If a relevant incident does already exist and is currently open, the alert is added to the incident as an additional *trigger alert*. An incident's trigger alerts are defined as any alert that either caused the incident to be created, or would have caused creation had an incident not already existed. An incident takes the name of its first trigger alert.

When an incident is created, any alerts sharing the same cause as the incident's trigger alerts are automatically added as *contextual alerts*. This list is kept up to date as subsequent related alerts are detected.

In a incident overview (or a in a ChatOps notification), you will also find alerts that are ranked as *probable cause alerts* by Cloud Pak for AIOps. That is, the alerts that are most likely to be the probable cause of the incident. The distinction between *probable cause alerts* and *trigger alerts* is that the trigger alert(s) indicated that the problem is severe enough to warrant an incident (and therefore triggered its creation), but they did not necessarily cause the issue the incident represents. The

probable cause alert (which could be a contextual alert) might have caused the incident, as it caused the underlying issue. Without the trigger alert(s), the incident is not created, but other parts of the issue (including the cause) might still be there.

The state of the incident is linked to the state of the incident's trigger alerts. If all trigger alerts within an incident enter the **`clear`** state, the incident will enter the **`resolved`** state, and, after 2 minutes, will close. Likewise, an incident can become unresolved (that is, go back to in progress) if any of the triggered alerts become open again. This can happen when a new problem alert is received before the incident was closed.

Incidents

The **Incidents** tab displays information such as incident's priority, status, description, and when it was opened. From here, you can change the status or priority of an incident, and also manage the assignees. Drill down into any incident in the list to see the probable cause alerts, alert topology, and any associated automations. When you click an incident in the table, you can see more details in the side panel.

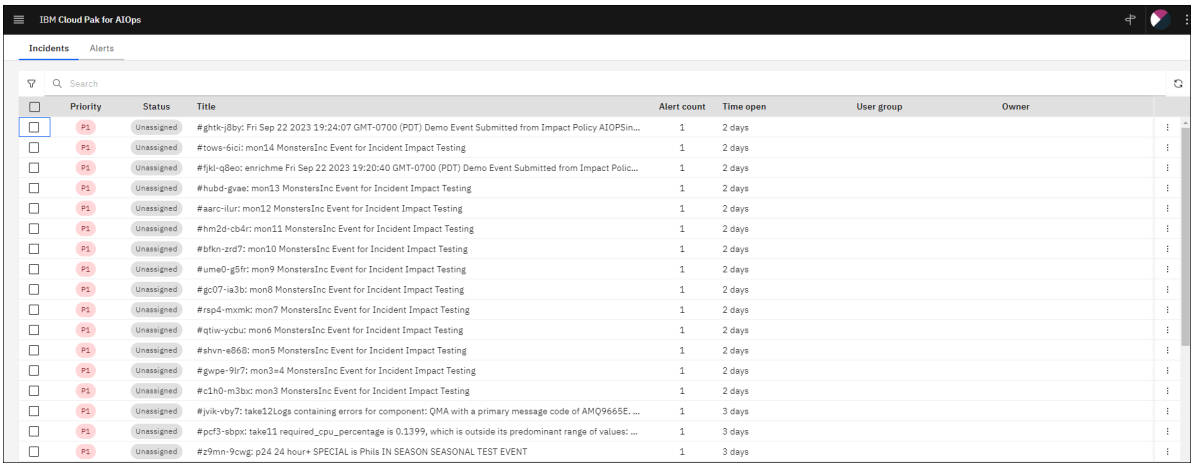


Figure. Incident UI

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Incidents**.
    - Click the menu overflow icon of any incident (the three dots at the end of the row) to **Change priority** (from P1 - highest priority to P5 - lowest), **Change status** (Unassigned, In progress, On hold, or Resolved), or **Manage assignees**.
    - Click on an incident's **Title** text to open the incident's **Overview** page.

Side panel

Click in the row of an incident you are interested in to open a **side panel** on the right-hand side of the table. It shows the incident's **Priority** (Priority 1 - highest priority through to Priority 5 - lowest) and current **Status** (Unassigned, In progress, On hold, or Resolved). You can click the options menu at the top of the side panel (three vertical dots) to change the incident's status, priority, or manage the assignees.

The **Incident ID**, the linked **ServiceNow** incident number, the linked **Slack** message and Slack reactive channel name, and when the incident was **Opened** are all displayed in the side panel. Click **Investigate** to go to the incident **Overview** page.

If the incident is assigned, the side panel shows the **Group** and **Owner** on the incident. If the incident is not yet assigned, click **Assign** to assign a group or owner. To change a current assignment, click **Edit** .

Applications that are impacted by the alerts that are part of the incident are listed under **Impacted applications**. The number of **Active incidents** that the application is impacted by is displayed. These incidents are categorized by **Priority level**. The importance of each impacted application to the business is also shown by a **Business criticality** indicator on the right of the

side panel. Applications are sorted based on priority. You can click an application name to launch out to the application in **Resource management**.

The following table lists the columns present in the **Incidents** tab.

Column	Description
Priority	Indicates the impact level of an incident on customer. Priority levels range from **P1** (highest priority) to **P5** (lowest).
Status	An incident can have a status of **Unassigned**, **Assigned**, **On hold**, **In progress**, or **Resolved**.
Description	A description of the incident. Hover the cursor over the text in the **Description** column to display a link to the Incident overview.
Time open	Time period since the incident was created.
User group	Group assigned to the incident.
Owner	User assigned to the incident.

The following icons are used on the incident list:

Icon	Name	Description
	Search	Click this icon to search for text in any of the columns displayed in the table.
	Filter	Click the filter icon to filter the incidents in the table based on any of the following criteria: • Priority • Status • User groups • Owner • Time open
	Refresh	Refresh the incident list to view all the latest incidents at the current point in time.

For more information about incidents in Cloud Pak for AIOps, see the following topics:

- Incident overview
- Searching and filtering incidents
- Managing multiple incidents

Incident overview

The incident **Overview** functions as a report on the best course of action by presenting the top probable causes, as well as similar past resolution tickets and associated automations to remediate the alerts. From a ChatOps integration (Slack or Microsoft Teams), you can launch into an incident's overview from the incident title link in the notification.

Click the **Alerts** tab for a list of all the alerts that are associated to the incident.

The **Topology** tab shows the relationships of impacted resources.

The **Activity** tab is timeline of major events that have occurred in the lifecycle of an incident. The following events are logged to the Activity timeline:

- When an incident is created.
- Incident is assigned to new owner or team.
- An alert has been associated with the incident.
- Incident state changes. For example, the incident has been set to **On hold** from **In progress**, or marked as **Resolved**.
- User added comment.

To add a comment to the incident timeline, type a comment in the field that is provided and click **+** Add comment. A maximum of 250 characters are supported. The comment is stored in the timeline in chronological order with the other entries.

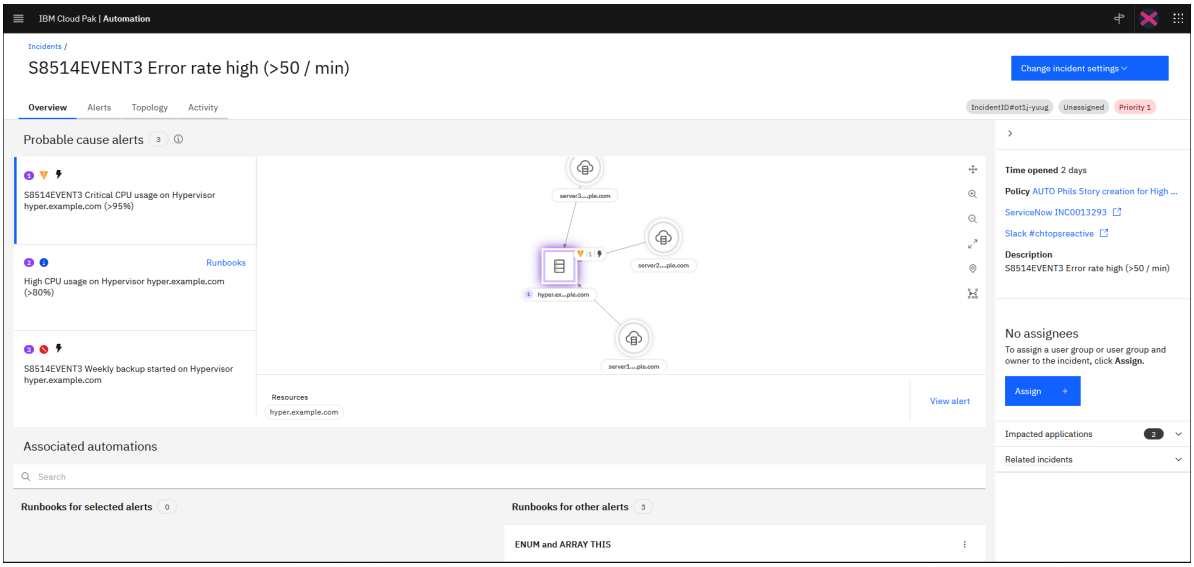


Figure. Incident overview

From within Cloud Pak for AIOps, complete the following steps to navigate to an incident's overview page.

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Incidents and alerts**.
3. From the **Incidents** tab, select an incident from the list and click the text in the **Title** column.

The following sections are presented on the incident **Overview** page:

- Probable cause alerts
- Alert topology
- Collapsible side panel
- Associated automations
- Similar past resolution tickets

For more information about the **Alerts** tab and **Topology** tab, see Alert Viewer and Viewing a topology.

Probable cause alerts

Based on the topology associated to an incident, and classification of the alerts, the alerts that are most likely to be the probable cause of the incident are presented here. Probable cause alerts show the top three most likely cause alerts, ranked in order of probability of being the cause of the incident.

The following information is displayed for each probable cause alert:

- **Probable cause ranking** 1 2 3

  Probable cause assigns a ranking to all of the alerts in the incident. Alerts are ranked in order of likelihood of being the cause of the incident.

- **Severity indicator**

  Indicates the severity of the alert. The following severity icons are used:

  - Critical
  - Major
  - Minor

- Warning
- Informational
- Indeterminate

- **Trigger alert**

  Denotes alerts that are defined as trigger alerts. That is, an alert that either caused the incident to be created, or would have caused creation had an incident not existed.

- **Summary**

  A description of the alert.

Note: Where probable cause cannot be identified, the section **Top Alerts** will appear instead, listing the top 3 alerts for the incident ranked in order by severity, first occurrence, trigger alerts, and finally by the availability of associated automations.

Select any of the probable cause or top alerts to see topology information for that alert.

To get more details about a probable cause alert, click **View alert** below the topology diagram. This brings you to the Alert Viewer, with the relevant alert selected and the **Alert details** displayed.

Alert topology

Select a probable cause alert (or top alert) from the list. If the resource on which the alert occurred can be located in the network topology system, a one hop topology map is diplayed for this alert, centered on the resource on which the alert occurred.

Restriction: In order for an incident topology to be generated from the alert, at least one topology group enabled for correlation **must** have been added to the alert.

- To investigate the incident topology, click **View topology** below the topology diagram. If the topology group condition is met, then this action opens the **Topology** tab, which provides a holistic view of the relationships between different infrastructure and application components.

For each resource associated with the incident, you can see the number of alerts, severity indicator, and probable cause ranking. This information can be used to rapidly identify the root cause and the impact of the issue on your environment before taking action.

Side panel

The side panel is displayed by default on the Incident overview page. When available, it shows the linked **ServiceNow** incident number, the linked **Slack** message and Slack reactive channel name, and when the incident was **Opened**. The **Policy** is the Cloud Pak for AIOps policy that created the incident. Click the policy name to locate that policy on the Policies UI.

Just above the Side panel, you can see the **Incident ID**, its current **Status** (Unassigned, In progress, On hold, or Resolved), and **Priority** (Priority 1 - highest priority through to Priority 5 - lowest). Click **Change Incident settings** to change the incident's status, change its priority, or manage the assignees.

If the incident is assigned, the side panel shows the **Group** and **Owner** on the incident. If the incident is not yet assigned, click **Assign** to assign a group or owner. To change a current assignment, click **Edit** .

Applications that are impacted by the alerts that are part of the incident are listed under **Impacted applications**. A number indicates how many applications have been impacted. Click the arrow to expand the list of applications. The importance of each impacted application to the business is shown by a **Business criticality** indicator on the right of the side panel. The number of **Active incidents** that the application is impacted by is also displayed. These incidents are categorized by **Priority level**. Applications are sorted based on priority. You can click an application name to launch out to the application in **Resource management**. If there are more than five applications, you can click **View all** to open a modal containing the complete list. You can sort the applications in the table by clicking on each column header within the modal table. To filter the list of impacted applications, enter a search string in the **Search** box within the modal and start typing the name of the application you are looking for. As you type, the list displays only applications that contain the typed text. As with the applications displayed in the side panel, clicking on an application's name opens that application.

Any incidents that share contextual alerts are listed in the **Related incidents** section of the side panel. Related incidents can provide additional context when you are resolving an incident. A number indicates how many related incidents have been found. Click the arrow ⌄ to expand the list of incidents. The priority and status of each incident is displayed, and the number of associated alerts. Related incidents are sorted based on priority. You can click an incident's title text to go to the incident's **Overview** page. If there are more than five related incidents, you can click **View all** to open a modal containing a complete list. You can sort the incidents in the table by priority, status, and alert count. To filter the list of incidents, enter a search string in the **Search** 🔍 box within the modal and start typing the name of the incident you are looking for. As you type, the list displays only incidents that contain the typed text. As with the incidents displayed in the side panel, clicking the title text opens that incident's **Overview** page.

To close the side panel, click the arrow › below the **IncidentID**.

Associated automations

The runbooks found for an incident's probable cause alerts or top alerts are listed under **Associated automations**. Select an alert to display the associated runbooks.

To search the list of runbooks, click anywhere in the **Search** 🔍 field and start typing the name of the runbook you are looking for. As you type, the list displays only runbooks that contain the typed text.

The following information is displayed for each recommended runbook:

- **Status**

  Successfully executed, Failed, Cancelled, In progress, or Completed.

- **Success rate**

  The success rate is calculated by using the number of successful and unsuccessful executions of the runbook.

- **Average rating**

  Operations analysts can provide feedback about the quality of a runbook including a rating, which is displayed here (five stars is the top rating).

- **Type**

  Indicates the Runbook type: Manual or Automated.

- **Policy**

  Shows the Cloud Pak for AIOps policy that assigned the runbook to the selected alert. Click the policy name to locate that policy on the Policies UI.

Select a runbook and click the menu overflow icon (the three dots at the end of the row). Select **Preview** to find out what a runbook consists of, or select **Run** to run the runbook.

Similar past resolution tickets

It can be helpful to review details of similar past resolution tickets to help determine the best course of action. Insights are used to display any similar past resolution tickets that are associated with the incident, together with a description of each ticket. If there are more than three similar past resolution tickets, click **View all** to open a modal containing a more complete list. You can also search for additional past resolution tickets using the modal's built in **Search** 🔍 bar. The search function queries the similar ticket service API to look for other **ServiceNow** tickets based on the search term.

Click the title of a past resolution ticket to launch out to the linked **ServiceNow** ticket.

See the following topics for more information:

- About events, alerts and incidents
- Viewing a topology
- Runbooks
- Policies

Searching and filtering incidents

Search for incidents by name, or filter the list based on selected critria.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Incidents**.

- To search for an incident in the table:
  - Click anywhere in the **Search** field.
  - Start typing a description of the incident in the search field. For example, type "failed" and the system will show in the table all incidents that contain the text "failed" in the description.
- To filter the list of incidents in the table:
  - Click **Filter** to open the filter side panel.
  - Select from the following filters for incidents:
    - Priority: P1 - highest priority, through to P5 - lowest.
    - Status: Unassigned, Assigned, In progress, On hold, or Resolved.
    - User groups: Find user groups to filter by.
    - Owner: Find owners to filter by.
    - Time open: under 1 hour, 1 hour to 1 day, 1 day to 7 days, 7 days to 15 days, 15 days to 30 days, 30 days and above.
  - Click **Done** to close the Filter window.

3. To clear all filters, click **Filter** and then **Clear all**.

Managing multiple incidents

You can multiselect incidents and apply actions to all of your selections at one time.

Procedure

1. Select multiple incidents from the table by using the checkboxes at the beginning of the table rows.

   Tip: You can select all of the incidents in the table by clicking the checkbox in the table header beside **Priority**.

2. A banner above the table displays the number of incidents that are selected, together with a menu of options. The following actions can be applied to multiple incidents at the same time:

   - Change priority: select from a range of **P1** (highest priority) to **P5** (lowest).
   - Change status: to **In progress**, **On hold**, or **Resolved**.
   - Manage assignees: assign a **Group**, **Owner**, and **Collaborators** to the incidents.

Alert Viewer

Alerts are displayed in the Alert Viewer. From a ChatOps integration (Slack or Microsoft Teams), you can launch into an Alert Viewer that shows the alerts that are associated to a specific incident. The numbers on the columns represent sort order. By default, alerts are sorted in the table by **Severity**. Where severity is the same, alerts are sorted by **First occurrence**. Each increasing sort order only sorts equivalent values. So once all values in a column sort order are unique (not equivalent), subsequent sort orders are ignored.

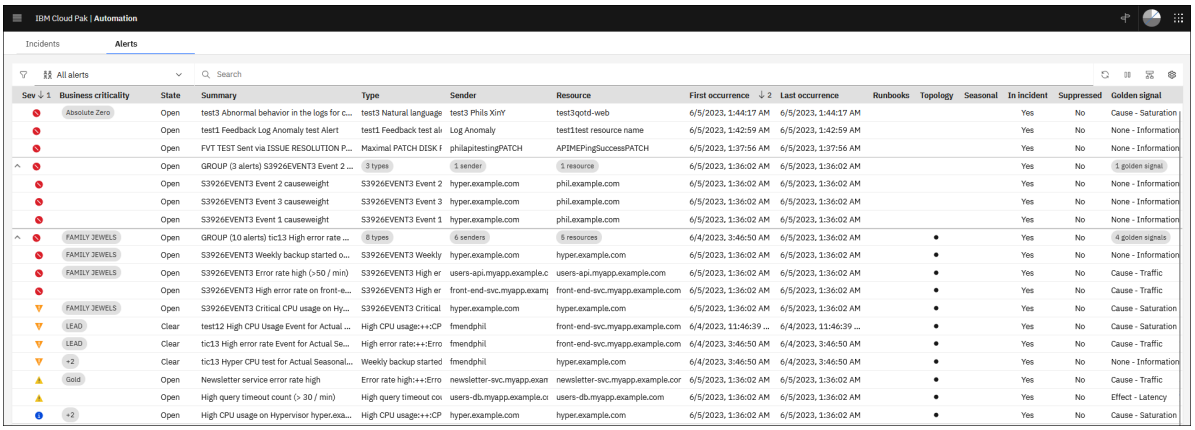


Figure. Alert viewer

There is also an alert list that displays all of the alerts found by IBM Cloud Pak® for AIOps. Follow the steps below to navigate to the main Alert Viewer.

Viewing alerts

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Alerts**.

The following table lists alert attributes.

Table. Alert attributes

Attribute	Description
Severity	Displays the severity of the alert. The possible values that can display in the Severity column are 6: Critical, 5: Major, 4: Minor, 3: Warning, 2: Informational, 1: Indeterminate.
Business criticality	Set in **Resource management** to define the importance of an application, resource group, or resource to the business.
State	An alert has one of the following three states: **Open** - Always has a severity other than `clear` to indicate an active issue that can require your attention. **Clear** - Working as expected, also has a severity of `clear`. **Closed** - No longer active or relevant, a deleted event. Also has a severity of `clear`.
Ranking	Probable cause assigns a ranking to all of the alerts in the incident. Alerts are ranked in order of likelihood of being the cause of the incident.
Summary	Provides more detail about the alert. For possible log anomaly summaries, see Log anomaly summaries.
Type	Describes the type of alert, for example, Utilization, System status, or Threshold breach. For possible log anomaly alert types, see Log anomaly alert types.
Sender	Identifies what, such as a log anomaly, that sent the alert. Different senders can result in different alert attributes.
Resource	The source from which the alert originated. This attribute can be a device name or hostname, service name, or application.
First occurrence	Displays the date and time in which the alert first occurred. The date format shown is based on your browser locale settings. Refer to the browser help for instructions on display format for dates, times, and numbers.
Last occurrence	Displays the date and time in which the alert last occurred. The date format shown is based on your browser locale settings. Refer to the browser help for instructions on display format for dates, times, and numbers.
Runbooks	A number in this column indicates the number of runbooks that are associated to an alert. Click the number to display the runbook details in the side panel. From here, you can see more information, preview, and run the runbook.
Topology	If the resource on which an alert occurred can be located in the network topology system, a large dot ● is presented in the alert's Topology column. Click the ● to display a topology map for this alert, centered on the resource on which the alert occurred.

Attribute	Description
Seasonal	Alerts that occur within a seasonal time window. Alerts that are not part of an alert group, that have associated seasonality, have a circle ○ visible in this column. Alerts that are part of an alert group, that have associated seasonality, have a dot ● in this column. Click the circle or dot to display more information in the **Seasonality** section of the **Alert details** panel.
In incident	Indicates if the alert is associated to an incident.
Trigger *	Denotes alerts that are defined as trigger alerts. That is, an alert that either caused the incident to be created, or would have caused creation had an incident not already existed. An incident takes the name of its trigger alert.
Suppressed	A 'Yes' in this column indicates that the `alert.suppressed` flag is set to `true` for the alert. The alert is still present in the system and can be viewed in the Alert Viewer, but the alert is filtered out of the view by default.
Golden signal	A label based on the value of `alert.type.classification` that describes the type of the event. For example, utilization, system status, threshold breach, and so on. Golden signal labels are divided into the categories of **Effect**: latency, error, availability. **Cause**: saturation, traffic. **None**: information.

*Only displayed on alert lists within the context of an incident.

Click the **Correlation information** icon to display three more columns. If the alert is part of an alert group, the type of analysis that is used to generate the group is indicated in these columns.

Table. Alert groups

Column	Description
Temporal group	Details of a temporal group in which this alert is involved.
Scope-based group	Details of a scope-based group in which this alert is involved.
Topological group	Details of a topology group in which this alert is involved.

For more information, see Displaying analytics details for an alert group.

Icons used in the Alert Viewer

Use this information to understand the purpose of the different icons used in the Alert Viewer.

Table. Icons used in the Alert Viewer

Icon	Name	Description
	Critical alert	A service-affecting condition has occurred, and corrective action is immediately required. For example, a system or device has gone out of service and needs to be restored.
	Major alert	A service-affecting problem occurred. Corrective action is urgently required. For example, a severe degradation occurred in the capability of a system or device and full capability must be restored.
	Minor alert	A non-service affecting problem occurred; take corrective action to prevent alerts of higher severity. For example, a problem occurred on a system or device but it does not impair the capacity or performance of the device.
	Warning alert	Potential or impending problems were detected. Further investigation is needed to prevent alerts of higher severity.
	Information alert	Alerts that are for information purposes only. Further investigation might be needed.
	Indeterminate alert	The severity level cannot be determined.
	Search	Click this icon to search for text in any of the columns displayed in the table.
	Correlation information *	Click this icon to display alert grouping information.
	Filter	Click this icon to filter the alerts in the table based on one of the following criteria: severity, enrichment, grouping, state, or alert type.
	Refresh	Refresh the Alert Viewer to view all the latest alerts at the current point in time.

Icon	Name	Description
	Temporal *	Based on alert history, these alerts tend to occur within a short time of each other.
	Scope-based *	Scope-based alerts occur within a configurable time window on an administrator-defined scope, such as a location, service, or resource.
	Topological *	Topological alerts occur on resources within a predefined section of your network topology.
	Trigger alert *	Denotes alerts that are defined as trigger alerts. That is, an alert that either caused the incident to be created, or would have caused creation had an incident not existed.
	Seasonal alert	Denotes a seasonal occurrence of an alert.
	Seasonal alert	Denotes a non-seasonal occurrence of a seasonal alert.

*Displayed only on alert lists associated to an incident.

Alert details

The **Alert details** side panel is accessed in the Alert Viewer. From here, you can quickly clear alerts or access additional information about the alert properties.

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. For the complete list of Cloud Pak for AIOps alerts, click **Operate** > **Alerts**.

   For the alerts associated to a specific incident, click **Operate** > **Incidents** > ***Incident Title*** > **Alerts**.
3. Select an alert in the table to display the **Alert details** side panel for that alert. The **Information** section is open by default in the side panel.

   Tip: To clear the selected alert, scroll to the end of the side panel and click **Clear alert** and then click **Clear** to confirm.

   - To display **Alert details** in a new window, click the pop-up icon in the side panel.
   - BETA *Providing feedback for improving training* is available only for technology preview, and is not for production usage.

   For some alerts, you can provide feedback to improve training a machine learning model.

   - Select **Alert is valid** if the alert is a genuine anomaly.
   - Select **Alert is not valid** if the alert doesn't indicate an anomaly.
   - After you make your selection, a corresponding message appears in **Alert details**. To revoke the selection, click **Undo**.
   - For more information, see **Alert insights** in Policy editor.
   - Selecting whether an alert is valid or invalid is available only for nonproduction use (technology preview).

Depending on the alert selected, **Alert details** can contain the following sections:

- Information
- Activity
- Anomaly insights
- Runbooks
- Topology
- Log anomaly
- Metric anomaly
- Recommended actions

Information

Under Information, alert properties can be viewed in two different formats:

- **Tabular**

Alert properties are displayed in a table format of **Property** - **Value** pairs.

- **Raw**

  Raw data in a JSON format. Click **Copy** to copy the alert data to the clipboard.

Activity

A timeline of major events that have occurred in the lifecycle of an alert. The following events are logged to the Activity timeline:

- When an alert is created
- Alert is assigned to new owner or team
- Alert is acknowledged/de-acknowledged
- Alert is suppressed/de-suppressed
- Alert state changes (open or clear)
- Alert severity changes
- User added comment

To add a comment to an alert at any time, type a comment in the field that is provided and click **+** Add comment. A maximum of 250 characters are supported. The comment is stored in the timeline in chronological order with the other entries.

Anomaly insights

In the Alerts table, if you select an alert that contains anomaly insights, the Alert details panel contains an Anomaly insights section. This section shows Property, which is the Key Performance Indicator (KPI) type, and Value, which is the name, for each anomaly insight. Use this information to understand the anomaly and why an alert was generated. You can also use this information to create a policy. For example, you can suppress the alert or associate it with a runbook. For more information, see Supported Anomaly insights properties.

Runbooks

The following information is displayed for each runbook in **Alert details**:

- **Status**

  Successfully executed, Failed, Cancelled, In progress, or Completed.

- **Success rate**

  The success rate is calculated by using the number of successful and unsuccessful executions of the runbook.

- **Average rating**

  Operations analysts can provide feedback about the quality of a runbook including a rating, which is displayed here (five stars is the top rating).

- **Type**

  Indicates the Runbook type: Manual or Automated.

- **Policy**

  Shows the Cloud Pak for AIOps policy that assigned the runbook to the selected alert. Click the policy name to locate that policy on the Policies UI.

Click the menu overflow icon (the three dots) to **View** what a runbook consists of, or to **Re-run** the runbook.

Topology

If the resource on which an alert occurred can be located in the network topology system, you can collapse the **Information** section in the side panel to view the **Topology** section. For more information, see Displaying alert topology.

Log anomaly

When you select a log anomaly alert in the table, the **Alert details** panel contains a **Log anomaly details** section. For more information, see Log anomaly alerts.

Domain-specific log anomaly alert details

Select a domain-specific log anomaly alert, such as one pertaining to IBM MQ or WebSphere, in the **Alerts** table. The **Alert details** panel contains a **Log anomaly** details section with the following information:

Table. Alert detail explanation

Detail	Explanation
Message code	Every domain-specific log includes a message code, and message codes are summarized in an alert. This message code is the most significant domain-specific message code for the current alert.
Short explanation	The title of the log anomaly is based on the message code.
Detailed explanation	The description of the log anomaly is based on the message code.
Category and Subcategory	The log anomaly is assigned to a category and subcategory based on the message code.
Message codes and frequencies	A log anomaly alert consists of multiple logs, and each log has a message code. The frequency shows how often a certain message code appears in the logs that make up the log anomaly alert.

Metric anomaly

When a metric anomaly alert is selected in the table, the **Alert details** panel also contains a **Metric anomaly details** section. For more information, see Metric anomaly details.

Recommended actions

For domain-specific log anomaly alerts, up to three links for resolving the alert are included. Each link consists of a title and a short description of the resolution. These links are based on the message code and past resolution tickets that are associated with the selected alert.

Correlation information

If you are viewing an incident's alert list, and the selected alert is part of an alert group, the Alert details side panel displays information about the underlying temporal, topological, or scope-based groups.

Working with alerts

Actions for alerts can be accessed via the right-click context menu.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Alerts**.
3. Right-click any alert (or multiple alerts) in the Alert Viewer and select from the following actions:

- **Acknowledge**

  Acknowledge an alert when you want to work on that alert. The acknowledged status is recorded as **`true`** in the Alert details.

- **De-acknowledge**

  De-acknowledge an alert if you are no longer working on it. The acknowledged status is recorded as **`false`** in the Alert details.

- **Update severity**

  Select from the following alert severities:

  - Indeterminate

    - Informational
    - Warning
    - Minor
    - Major
    - Critical

- **Suppress**

  When an alert is suppressed in Cloud Pak for AIOps, the **`alert.suppressed`** flag is set to **`true`**. The alert can still be viewed in the Alert Viewer, but the alert is filtered out of the view by default.

- **De-suppress**

  Changes the **`alert.suppressed`** flag from **`true`** to **`false`**. The alert is not filtered out of the view by default.

- **Take ownership**

  Take ownership of an alert if you want to work on resolving that alert.

- **Assign owner**

  Use this command to assign an alert to another user. That user then becomes the alert owner. You must select a user group the owner belongs to.

- **Assign user group**

  Use this command to assign an alert to a group.

- **Clear**

  Change the alert state to Clear.

- **Quick Filter**

  Use quick filters to display alerts that match a selected criteria. For more information, see Creating quick filters.

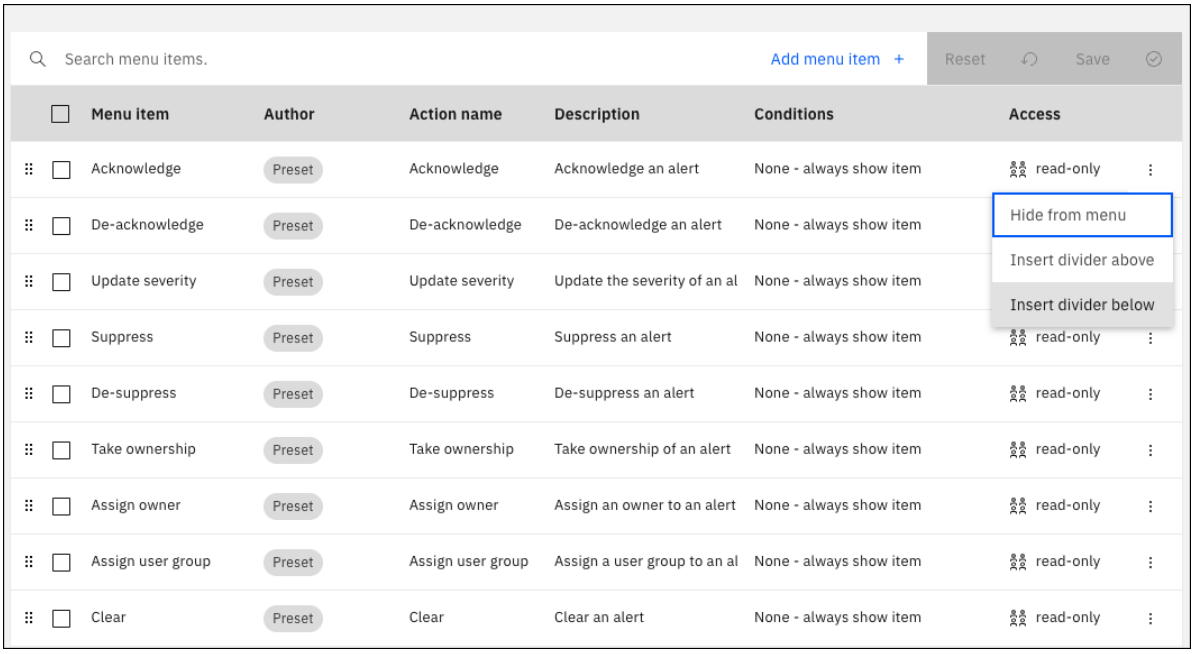
Menu customization

You can personalize the arrangement of right-click menu items. The arrangement is only visible to the user customizing the menu. That is, each user can define how their own menu looks. Users can control the following settings:

- Menu items are hidden or visible
- The order of menu items within the menu
- Grouping menu items with dividers

To access menu configuration in the Alert Viewer, click > **Configure alerts contextual menu**.

The menu configuration page represents the layout of the menu. From here, you can click and drag menu items to re-arrange the order. You can also click the options menu (three vertical dots) to hide or show menu items, and group menu items with dividers.



Menu configuration page

If a user has access to shared menu items, initially these items are shown as disabled in the menu configuration page. To enable shared menu items, select **Show in menu** from the options menu.

Add menu items

To add custom menu items for use in your personal menu, or to share with everyone, click **Add menu item +** from the menu configuration page.

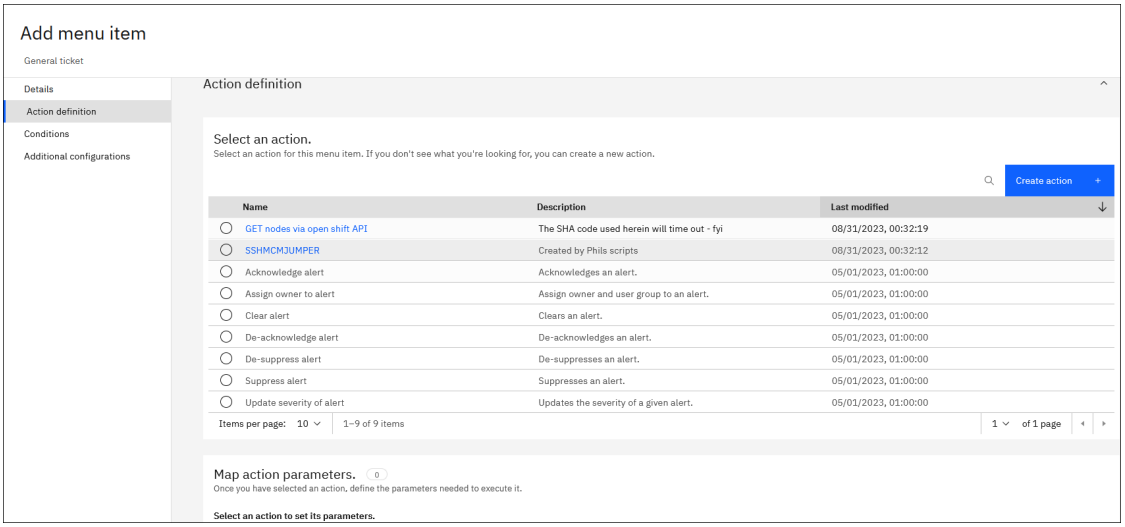
Configure the following properties for a custom menu item:

- **Details**
  - Display name
  - Description
  - Select who can use this menu item (**Only me** or **Everyone**).

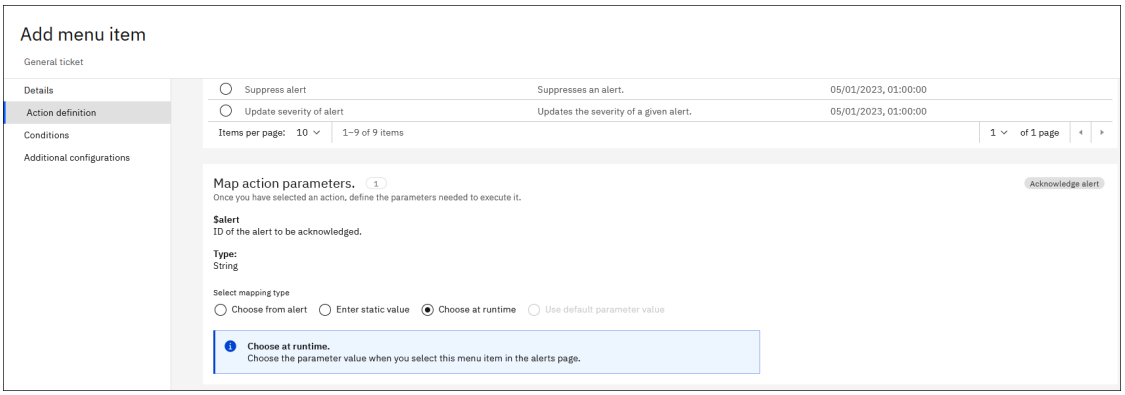


Add menu item - Details

- **Action definition** and parameter-mapping that includes the following options:
  - Alert properties: based on the selected alert(s), set the desired properties as the action parameter value.
  - Static value: set a fixed value for a parameter.
  - Default value: use the default value specified in the Automation action definition.
  - Choose at run time: input the value at the time the menu item is run.

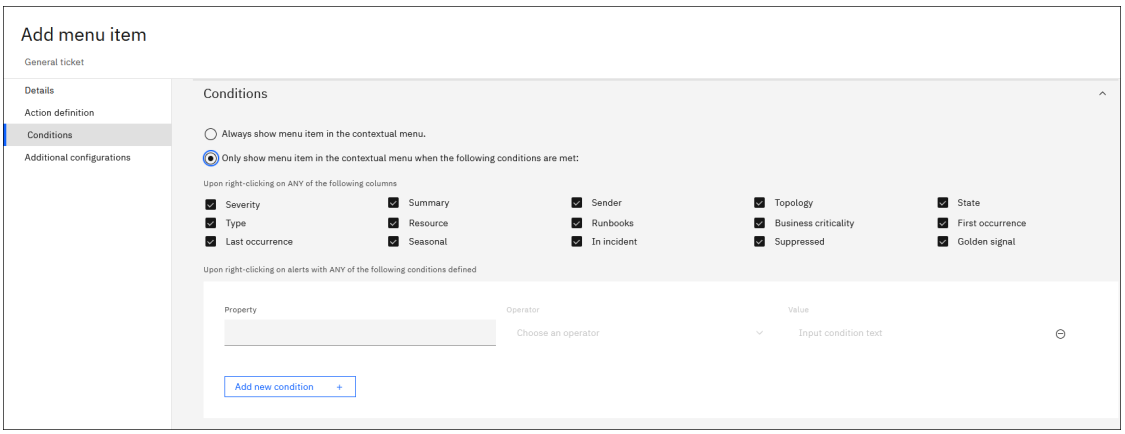


Add menu item - Action definition



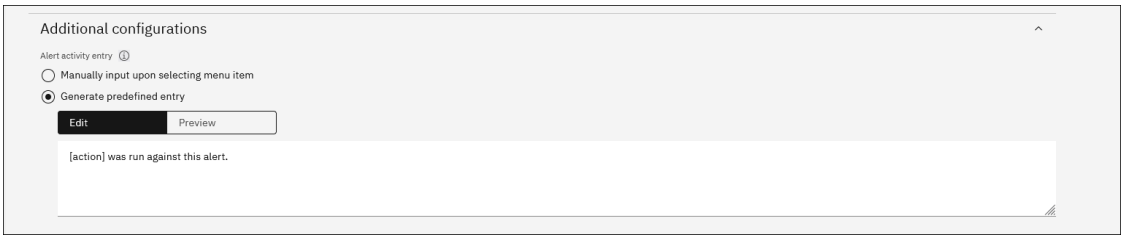
Add menu item - Map action parameters

- **Conditions** to define when a menu item should show on an alert right-click action:
  - Columns: select specific columns for which the menu item will surface.
  - Alert properties: select values of alert properties for which the menu item should show.



Add menu item - Conditions

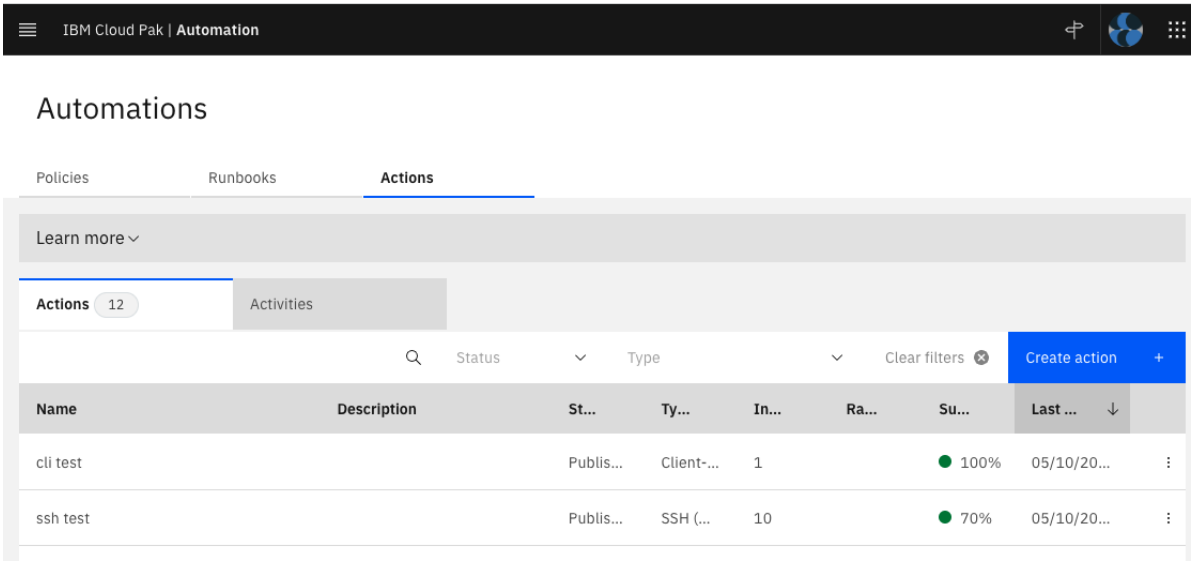
- **Additional configurations** to define the alert activity entry. This is the message that will appear in an alert's activity once this menu item has been selected to act on the alert. To change the default entry, edit the message and preview the output.



Add menu item - Additional configurations

Implementation

The actions that are associated with custom menu items are implemented as Automation Actions.



Automation Actions

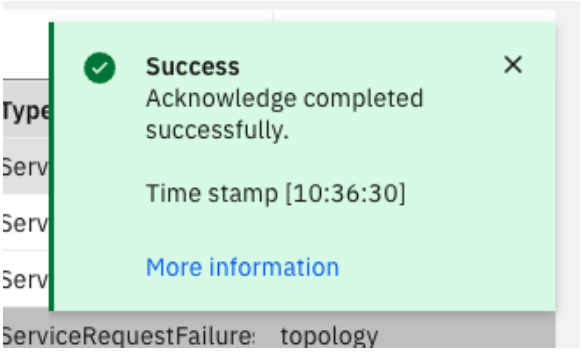
Actions are one of the following types:

- HTTP
- SSH
- Client-side

One benefit of using the Automation action framework for implementation is that these actions can be used independenly of the right-click menu, either stand-alone or within a runbook. For more information, see Managing actions.

Menu item execution

Upon clicking a menu item, the actions will run in the background for each selected alert if there are no parameter mappings specified as **Choose at run time**. When all actions are completed, a notification is displayed with results for each selected alert.



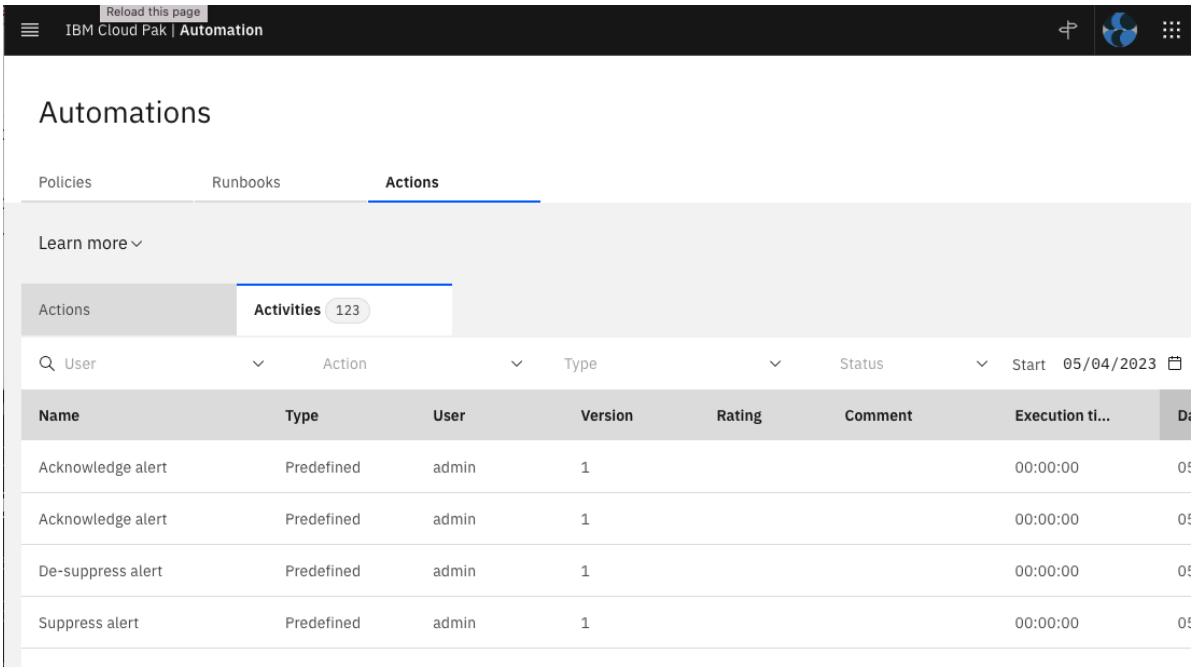
Menu item notification

Click the **More information** link in the notification to show the raw output from the action execution, which can be copied with the copy icon .



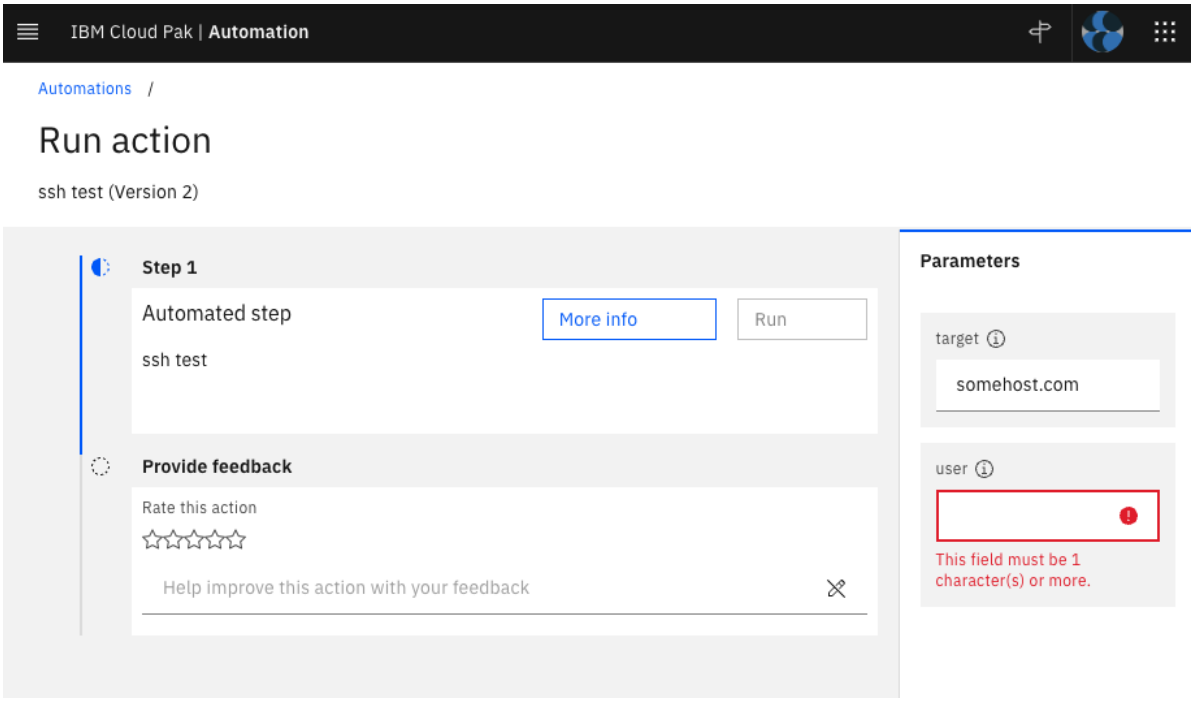
More information

An audit of all action executions is available on the Automations > Actions > **Activities** page.



Activities page

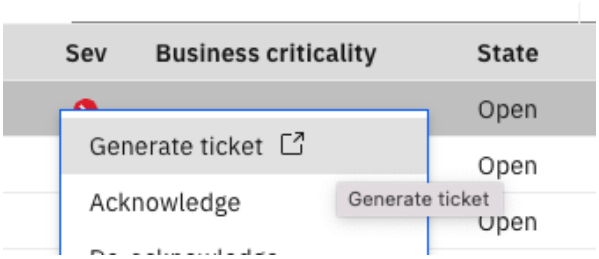
If a menu item has a parameter mapping specified as **Choose at run time**, the user is directed to the **Run action** page to complete the parameter input and run the action when clicking a menu item.



Run action

In this scenario, when multiple alerts are selected the alert properties are combined into a comma-delimited list. The automation action can then process this list as required.

For menu items using an action of type Client-side, if there are no parameter mappings of **Choose at run time**, the parameter values are mapped to URL query parameters and a new browser tab opens with the resolved URL when running the menu item. These menu items are differentiated by the "launch" icon. Note, menu items of this type only work for single alerts. If multiple alerts are selected, properties of the context menu alert are used to resolve parameter mappings.



Launch icon

Access control

Access control is applied at the menu item level. The menu items available for a user's personal menu consist of items the user has authored themselves, or items set with "Everyone" access. Preset menu items are read-only for everyone. Custom menu items can be updated by anyone with access to the menu item.

Searching and filtering alerts

Search for alerts by name, or filter the list based on selected criteria. You can also save a filter configuration for future use.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Alerts**.

Search alerts

To search for an alert in the table:

- Click anywhere in the **Search** field.
- Start typing a description of the alert in the search field. For example, type "error" and the system will show in the table all alerts that contain the text "error" in the summary.

Predefined alert filters

All Alerts are displayed in the alert list by default. Click the Down chevron icon to select from the following predefined filters:

- Alerts with runbooks
- Alerts with topology
- Critical
- Open alerts
- Part of an incident

Customized alert filters

1. Select an existing filter from the drop-down menu or use the default All Alerts.
2. Click the Filter icon to open the filter side panel.
3. Apply filters to the alert list (or apply additional criteria to an existing filter). Select from the following filters for alerts:
    - **Severity**:
        - Critical
        - Major
        - Minor
        - Warning
        - Informational
        - Indeterminate
    - **State**:
        - Open
        - Clear
        - Closed
    - **Insights**:
        - Part of an incident
        - Runbooks available
        - Enrichment
    - **Grouping insights**:
        - Temporal
        - Scope-based
        - Topological

- **Suppressed**:
  - Yes
  - No
- **Other properties**

  You can create more personalized filters by using alert properties. Expand the **Other properties** section and complete the fields as follows:

  - **Property**: select from alert attributes that are predefined for Cloud Pak for AIOps and common to most alerts.
  - **Operator**: select a comparison operator from this list. The range of comparisons available is determined by your selection from the Property drop-down list.
  - **Value**: the fields that appear here are dependent on the options that are selected in the Property and Operator lists.

To update an existing filter

1. After you have applied the filters, click **Save**.
2. A **Filter updated** message is displayed to confirm that your changes have been saved.

To save as a new filter

1. After you have applied the filters, click the three vertical dots in the **Filter conditions** side panel to open the list of options.
2. Click **Save as a new filter**.
3. Enter a **Filter name**, **Description**, and then select who can use this filter from the following options:
   - **Only me**
   - **Specified users, user groups, or both**: to manage access, you can select users, user groups, or both. Any selections you make includes yourself. After you select the users or user groups who can use the filter, you must specify their level of access:
     - **Can use**: users can see the filter in the drop-down list and apply it to the list of alerts.
     - **And edit**: users can use, edit, and save the filter.
     - **And manage**: users can use, edit, and manage the filter name, description, and access control.
   - **Everyone**: additionally specify if everyone can **Use this filter** or **Edit this filter**.

Note: A search term entered in the **Search** text field is not part of a saved filter. Only the configuration applied in the **Filter conditions** side panel is part of the saved filter.

To clear changes that have been applied to a filter, click **Reset filter** in the filter side panel or **Clear all** from the list of options (three vertical dots). From the options list you can also **Delete filter** and **Edit filter settings**.

Creating quick filters

You can use quick filters as a fast way of displaying alerts that match a selected criteria. For example, you can quickly display only those alerts that occurred at the same time as, or before, a selected alert. A quick filter is applied via the right-click context menu. Quick filters when applied are merged into the filter side panel together with the base filter selection.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Alerts**.
3. Right-click on any column or cell in the Alert Viewer, then select **Quick Filter** and choose from the following operators:

- Equals
- Does not equal
- Contains
- Does not contain
- Greater than
- Greater than or equal
- Less than
- Less than or equal

The **Insights** columns for **Part of an incident**, **Runbooks**, and **Topology** can be filtered using the **Contains** operator.

Results

The alerts are filtered to only those that meet the filter condition. The applied filter condition is displayed in the filter side panel.

Related links

Menu customization

You can personalize the arrangement of right-click menu items. The arrangement is only visible to the user customizing the menu. That is, each user can define how their own menu looks.

Setting user preferences for the Alert Viewer

You can configure the appearance and behavior of the Alert Viewer via the **User preferences**.

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Alerts**.
3. On the upper right corner of the page, click > **User preferences**.

   The following preferences can be applied to the Alert Viewer:

- **Default filter**

  Set a default filter for the alert list from the predefined or customized alert filters available in the drop-down list. For more information, see Searching and filtering alerts.

- **Row size**

  Select from the following table row sizes:

  - Very small
  - Small
  - Medium
  - Large

- **Alert details launch type**

  Selecting an alert in the table opens the alert details for that alert. These can be displayed in a **Side panel** or in a separate **Pop-up window**.

- **Row coloring**

  Change how alert severity is depicted in the table by applying a different color background for each severity level.

- **Table auto-refresh**

Enable auto-refresh to have the alert list automatically updated at predescribed intervals.

- **Table auto-refresh interval (s)**

  Auto-refresh interval in seconds.

4. Click **Save** to apply your changes to the Alert Viewer.

About alert groups

An alert group is a group of two or more alerts that Cloud Pak for AIOps has correlated together because the underlying analytics have determined that these alerts belong together.

Alerts can be added to an alert group because of one or more of these factors:

Correlation type	Example
Temporal groups: based on alert history, these alerts tend to occur within a short time of each other.	A "Latency" alert on a server is regularly followed by a "Ping response time high" alert on that same server. These alerts are grouped into a temporal subgroup.
Topological groups: the alerts occur on resources within a predefined section of your network topology.	If there is a predefined section of the network that groups together a specific switch and all the nodes that depend on that switch, then any alerts occurring on that specific switch or the nodes connected to it are grouped together. These alerts are grouped into a topological subgroup.
Scope-based groups: the alerts occur on a user defined scope.	An administrator defines a `ScopeID` based on properties that define the scope for how alerts are grouped. Any alerts that match the scope and occur within a default time window are then automatically grouped together. An example would be where an alert storm occurs on the `london145.acme.com` server. All of the alerts in that storm will be grouped together as they match the `alert.resource.location= london145.acme.com`, and they occur within the default time window. These alerts are grouped into a scope-based subgroup.

Displaying an alert group

Expand an alert group to display the alerts that were correlated together within the group.

An alert group contains two or more alerts, which are correlated together by the underlying analytics. The group can include **Temporal** groups, **Topological** groups, and **Scope-based** groups.

1. Identify an alert group within the table.

   You can identify a group using the following signs:

   - It has a Down chevron icon in its row in the table.
   - By default, it has a summary that includes the words: GROUP (n alerts), where n is the number of the alerts in the group.
2. Open the alert group by clicking the Down chevron icon in its row in the table.

   The group's alerts are now displayed under the parent alert.

Displaying alert topology

If the resource on which an alert occurred can be located in the network topology system, then a large dot ● is presented in the alert's Topology column. Click the large dot ● to display a topology map for this alert, centered on the resource on which the alert occurred.

Procedure

1. Click the large dot ● in the Topology column for the alert of interest.

   A side panel containing multiple information sections opens on the right side of the table. The Topology section is open and provides the following information:

   - **Resource**

     Name of the resource on which the alert of interest occurred.

   - **Begin time**

     Date and time when the topology for this resource was last updated.

   - **Observed time**

     Date and time when the alert of interest was observed on this resource.

   - **Topology pane**

     Shows a one hop topology map for this alert, centered on the resource on which the alert occurred.

2. To investigate the topology further, click the **More information** link at the bottom of the side panel's Topology section.

   This action opens the Topology viewer, which displays the same topology map, with the full topology display experience. For more information on how to use the Topology viewer to explore topology, see the related link at the bottom of this topic.

   Note: From an incident's Alert Viewer, the Topology viewer opens in the **Topology** tab on the same page. From the general Alert Viewer, the Topology viewer opens in a separate browser tab.

Related links

Viewing a topology

You can use the interactive Topology viewer tool to drill into the individual resource details and status, enable a timeline to view changes over time, and more.

Displaying alert seasonality

Alerts can occur within a seasonal time window. For example, an alert might tend to occur on a certain day of the week or a certain time of the day. Alerts that tend to occur within a seasonal time window are highlighted in the Alert Viewer using a circle ○ or a circle with a check mark ⊙ in the alert's **Seasonal** column. Click these icons to find out when the alert tends to occur.

Examples of seasonal time windows include the following times:

Hour of the day

```
Between 12:00 and 1:00 pm
```

Day of the week

`On Mondays`

Day of the month

`On the 3rd of the month`

Day of the week at a specific hour

`On Mondays, between 12:00 and 1:00 pm`

Day of the month at a specific hour

`On the 3rd of the month, between 12:00 and 1:00 pm`

Procedure

1. Identify alerts that are seasonal.

   - Alerts that are not part of an alert group, that have associated seasonality, have a circle ○ visible in the **Seasonal** column.
   - Alerts that are part of an alert group, that have associated seasonality, have a circle with a check mark ⓒ in the **Seasonal** column, but you must first open the alert group in order to see the alert. Open the alert group by clicking the Down chevron icon ⌄ in the parent alert group row.

2. Click the circle or dot to display more information on the alert of interest.

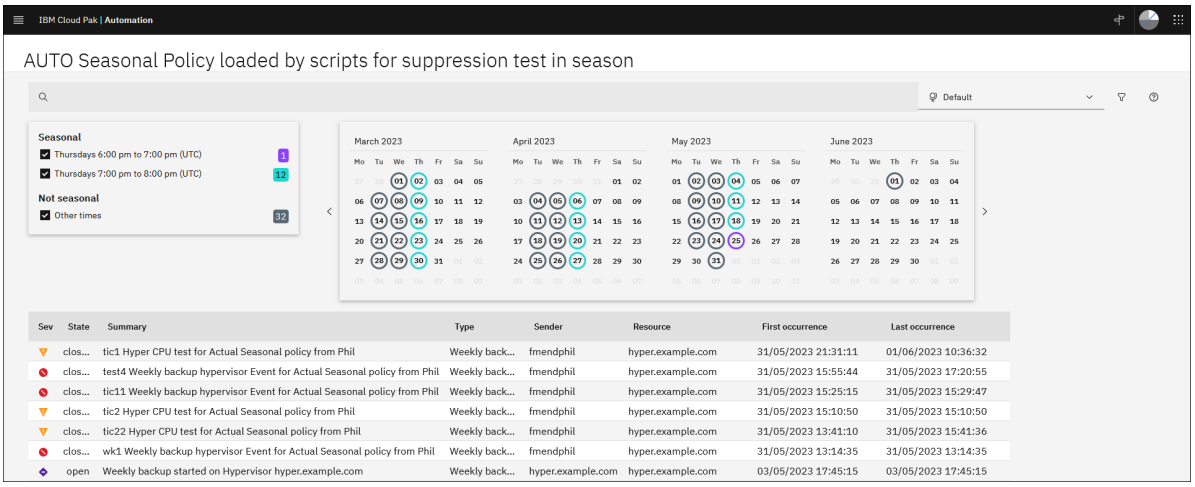
   A side panel that contains multiple information sections opens in the table. The **Seasonality** section is open and displays one or more seasonal time windows for the selected alert.

3. Click the Down chevron icon ⌄ next to each seasonal time window to see a visual representation of the time window.

   For example, if the underlying algorithm has identified that the selected alert tends to occur between 12 noon and 1:00 pm, then this alert will have a seasonal time window as follows, and this will be displayed using a clock-based representation:

   `Between 12:00 and 1:00 pm`

4. Click the **More information** link at the end of the side panel's **Seasonality** section to launch out to the **Seasonality details** page.



Seasonality details

This page provides a calendar view of all of the historical alerts that contributed to the selected seasonal alert. One or more seasonal time windows are listed in the top left pane. Select a seasonal time window to filter the view to show just historical alerts that contributed to that time window. The **Seasonality details** page contains the following sections:

- **Toolbar**
  - Search: searches alert data in all event group instances that are shown on this page.

- Filter: filters the alerts that are shown by severity level:
  - Critical
  - Major
  - Minor
  - Warning
  - Informational
  - Indeterminate

- **Seasonal time windows**

  One or more seasonal time windows are listed together with a number indicating how many historical alerts occurred within this time window during the historical period analyzed. The color of the square is the color code for the respective seasonal time window. Select a seasonal time window to filter the Calendar and the Historical alert table to show just historical alerts that contributed to that time window.

- **Calendar**

  A calendar of the historical period analyzed (the last three months or more) showing days on which the historical alerts of interest occurred, and colored with the color code for the respective time window. Click one or more days in the calendar view to filter the table of alerts details to the selected date.

- **Historical event table**

  Table listing the historical alerts during the historical period analyzed.

Displaying temporal correlation

The system continually analyzes past alerts to determine which alerts tend to frequently co-occur. When these alerts occur together again, they are correlated.

The temporal details page shows the alerts that make up a temporal group, together with all of the historical instances of the group. You can navigate into these instances to view detailed timelines of each instance. You can also view alert details.

Procedure

1. Identify alerts that are temporal in the Alert Viewer. You can click the Filter icon to filter by Grouping insights.
2. Click the **View correlation** icon to display the **Correlation** column in the table.
3. Clicking the temporal icon in this column opens the sidebar, with the **Temporal correlation** section open. This section contains the following information:

- **First group instance**

  Date and time of first instance of this group.

- **Total group instances**

  Total number of historical instances of this group. For details of when these instances occurred and how many alerts occurred in each instance, see the Group instance heatmap.

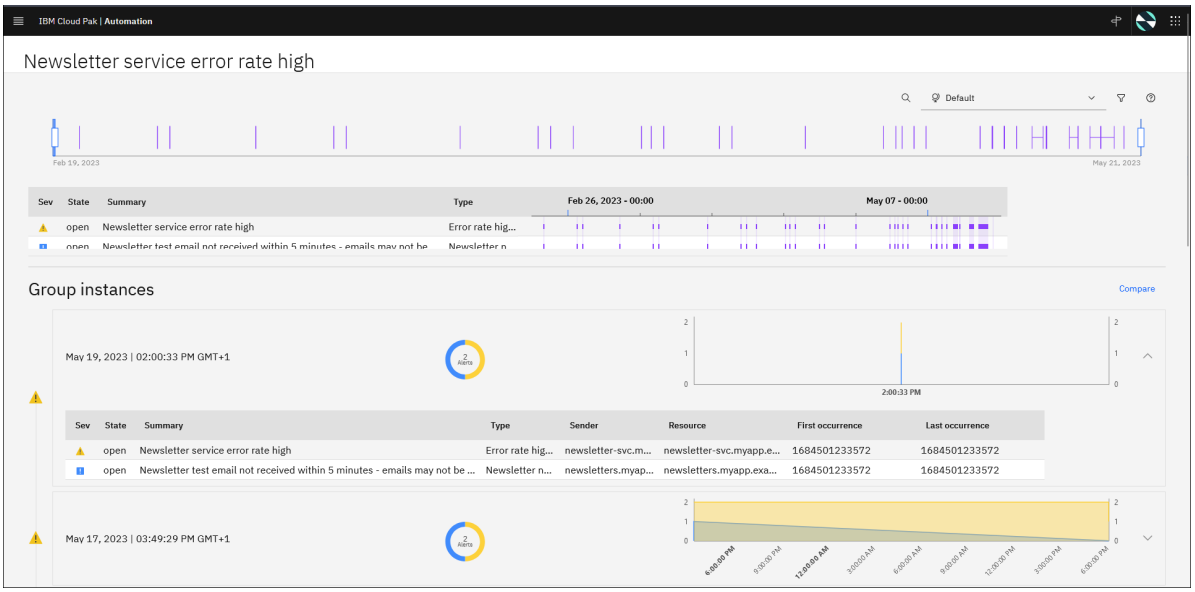
- **Average instance duration**

  Average time in seconds that this group instance lasted.

- **Group instance heatmap**

  Time-based heatmap showing recent historical period in days with a grey square for each day. Each darker square indicates a day on which there was at least one group instance. Hover over the square to see details of this group instance.

4. Click the **More information** link at the end of the side panel's **Temporal correlation** section to launch out to the temporal details page.



temporal correlation details

The temporal details page shows the alerts that make up a temporal group, together with all of the historical instances of the group. You can navigate into these instances to view detailed timelines of each instance. You can also view alert details. The temporal details page contains the following sections:

- **Toolbar**
  - Search: searches alert data in all event group instances that are shown on this page.
  - Filter: filters the alerts that are shown by severity level:
    - Critical
    - Major
    - Minor
    - Warning
    - Informational
    - Indeterminate

- **Overview timeline**

  Displays alert group instances over time and controls the display of alert group instance data on the rest of the page. By default the time range sliders are open sufficiently to show data on all alert group instances. Modify the time range by either clicking and dragging over the desired range inside the timeline, or by dragging the sliders to the desired range. The rest of the screen updates accordingly.

- **Alert group instance timeline**

  Displays all of the alerts that have historically participated in instances of this temporal alert group. The instance map provides a graphical view over time of when the various instances have occurred.

- **Alert group instance details**

Displays the following information for each alert group instance:

- Start date and time of event group instance: indicates the first occurrence value of the first alert in the alert group instance.
- Distribution of event severity values: pie chart providing a visual indication of the alert severity values. Hover over the pie chart for more details.
- Sparkline: chart of alert occurrence over time.
- Duration of event group instance: duration of the alert group instance, in text.
- Down chevron icon: click the down chevron icon ⌄ to see an alert table showing column details for each alert in this group instance.

Log anomaly alerts

Log anomaly alerts have a **Sender** attribute of **Log Anomaly**.

Types

For log anomaly alerts, the **Type** attribute shows which of the log anomaly models detected the anomaly:

- Natural language anomaly
- Statistical baseline anomaly

Summaries

The **Summary** attribute of an alert provides more detail about why an anomaly was detected.

A summary might look similar to the following example:

```
Logs containing errors for component: QMA with a primary message code of AMQ9665E. Evidence includes: domain-specific entities.
```

- The **Evidence includes** statement can show the following values:
  - `domain-specific entities`, for example, message codes
  - `embeddings (statistical baseline)`
  - `error entities`, for example, HTTP error codes and exceptions
  - `patterns + embeddings (natural language)`
- A **primary message code**, such as `AMQ9665E`, is included only for alerts where an anomaly is detected in logs that originate from a domain-specific source, such as IBM MQ or WebSphere.

Categorization of log anomaly alerts

During the training phase, templates are extracted based on the input logs. Models are trained to represent the normal behavior of a component or service in an incident-free window. In the inference, any incoming new live log data is matched against the templates that were extracted during training, and the distribution for a fixed window is computed. Any new logs, error logs, or deviation from the normal distribution from training might be flagged as an anomalous window and generate a log anomaly event.

Alerts that are generated for log anomalies come in two types:

- Alerts with the `Logs containing errors for` prefix are log anomaly alerts that are generated when an error template is included in the alert. Templates include internal lists of error keywords.
- Alerts with the `Abnormal behavior in the logs for` prefix are log anomaly alerts that are generated when no error templates are included in the alert.

A template is classified as an error by using keyword matching with a predefined error dictionary and error symptom words. For example, a template can be classified as **`Logs containing errors for`** when it includes the keywords **`expires`**, **`failure`**, or **`incomplete`**.

Explainability

An alert is generated when a log anomaly is detected. The metadata for an alert contains fields and information that can help you understand the identified log anomaly.

See the following alert field descriptions to better understand reported log anomalies:

- **`description`**: If error templates were found in the ten-second window and if the count of these templates is greater than **`0`**, these templates are captured in the **`description`** field. If error templates are not present after an internal ranking algorithm is applied, the top five templates that have a count value greater than **`0`** are included in this field.
- **`details`**: This field includes the **`text_dict`** field, which contains critical fields, such as **`template_ids`** and **`template_list`**. This field includes the sorted templates and entities based on an internal ranking algorithm that corresponds to the log anomaly. This ranking algorithm prioritizes error templates and low occurrence error templates. It also prioritizes error entities over other types of entities. Based on the ranking, only the top 5 templates are shown.

Microsoft Teams and Slack connections

If a Microsoft Teams or Slack connection is set up, the alerts are also sent to those connections. The same alert data is parsed and shown in Microsoft Teams or Slack.

- The Microsoft Teams or Slack UI parses and displays the templates that are associated with the alert.
- Use the **preview logs** feature to see a sample of the logs that are associated with a template.
- When you click **Attach log message patterns** on the Microsoft Teams or Slack UI, you receive all anomalous and normal logs in the ten-second window time range for the resource, also known as the application component, that is associated with the alert.
- For more information, see Configuring Slack applications for connection and Creating a Microsoft Teams connection.

High risk alerts from change risk

When a new, incoming change request is evaluated as high risk, an alert is posted to the IBM Cloud Pak® for AIOps Platform UI. It can be included in the Events report and can be grouped into an incident. By reporting these alerts, high risk insights can be used for incident avoidance and mitigation.

A change risk high risk alert can be identified in the Alert Viewer when the **Sender** column for an alert contains **`change-risk`**. So a high risk alert resembles this example:

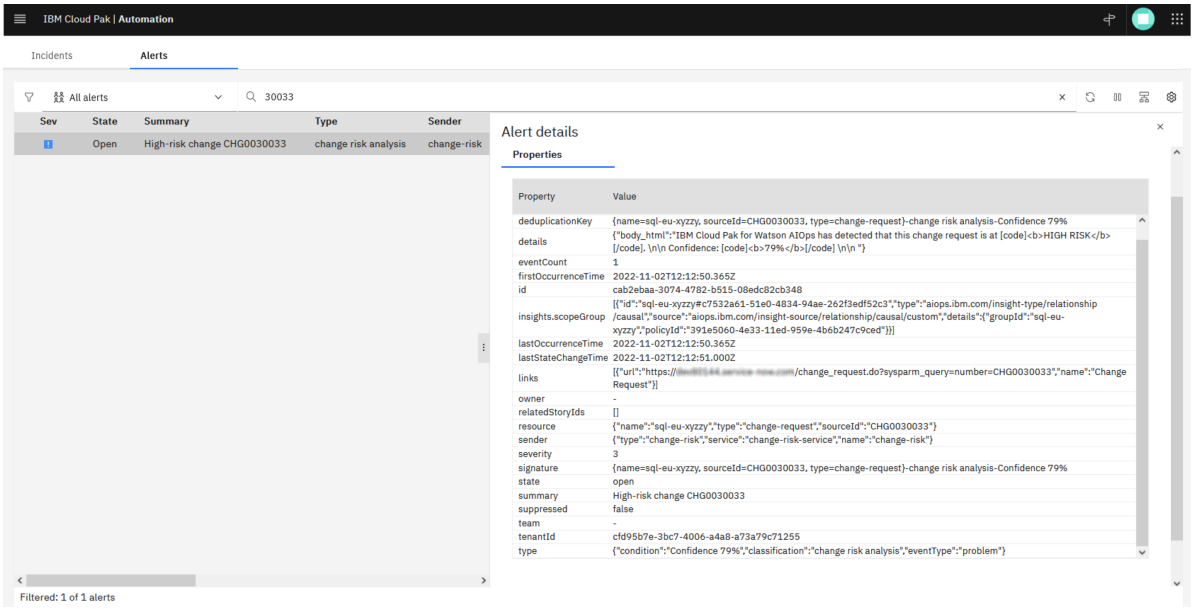


Figure. Alert viewer high risk

High risk in Events example:

```
id: UUID (str)
 sender: {
     service: "change-risk-service"
     name: "change-risk",
     type: "change-risk"
 }
 resource: {
     name: str, optional - Service Now CMDB ID (if present)
     sourceId: str - Change number (e.g. "CHG12345")
     type: "change-request"
 }
 type: {
     classification: "change risk analysis"
     condition: str - Confidence level (e.g. "Confidence 94%")
     eventType: "problem"
 }
 severity: 3 (int)
 summary: str - Change reference (e.g. "High-risk change CHG1")
 details: {
     body_html: str - HTML-marked English description of the change, related changes, and
related incidents.
 }
 occurrenceTime: datetime (str, UTC, ISO 8601 with timezone) - Time the risk evaluation was
processed
 expirySeconds: 0 (int)
 links: [
     {
         name: "Change Request",  (e.g. 'Change Request')
         url: str (e.g. 'https://example.service-now.com/change_request.do?
sysparm_query=number=CHG1')
     }
 ]
 insights: [] (empty list)
```

Viewing metric anomaly details

In **Incidents and alerts**, alerts can appear in the Alert Viewer table to indicate metric anomaly detection. More detailed information on these alerts, and a Key Performance Indicator (KPI) timeseries chart, can be viewed by opening the *Alert details* panel.

Note: A KPI is made up of a resource, which is a physical entity such as a network interface card, and a metric. For example, if a server has 2 network interface cards and you want to analyse data for 10 metrics on each network card in the server, the total number of KPIs is 20. KPIs are often interchangeably referred to as metrics, so a metric can sometimes refer to a measurement that is taken for a type of device - for example **`CPU_Context_Switches`**, and sometimes it can refer to an instance of that metric - for example **`CPU_Context_Switches`** on server **`ABC.acme.com`**.

Procedure

1. In the Alert Viewer (**Incidents and alerts** > **Alerts** tab) click anywhere on the row of a metric anomaly alert (identified by ANOMALY displayed in the *Type* column in the table).
2. An *Alert details* side panel opens. By default, the *Metric anomaly details* chart displays, showing the properties of the alert and associated values.
3. The chart plots the activity of the anomaly over a 2-day period (the two days up to the date in the date field).
4. To increase the viewing size of the chart within the panel, click the 3 vertical dots in the gray tab on the left side of the panel. Then, drag the tab to the left.

Understanding the chart

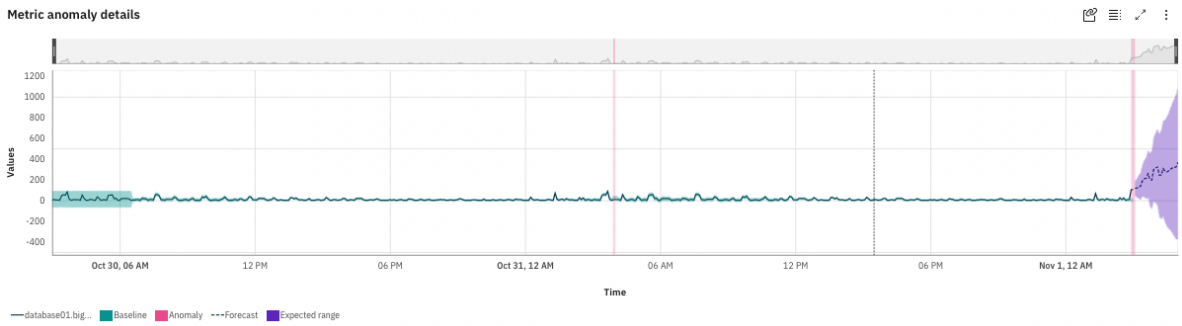


Figure. 2 Day chart

- The unbroken black line that is labeled with the name of the metric resource that is plotted on the chart represents the values of the selected metric over time.
- A green-shaded area (labeled *Baseline*) represents the acceptable operating range of that anomaly.

  Note: The default KPI Baseline is always that of the primary metric. To change to another Baseline, from related alerts or additional metrics, click the radio button (blank circle) attached to another KPI. Making this Baseline adjustment also changes the KPI that is forecast. If you select multiple metrics, you can choose to view only one Baseline from all of the KPIs that are in the related alerts and additional metrics tabs.
- A red-shaded area or red vertical line (labeled *Anomaly*) indicates where the metric is anomalous: one or more of the anomaly detection algorithms determined the metric was not behaving in accordance with the learned normal behavior.
- A purple area in the chart (labeled *Expected range*), with a dashed line (labeled *Forecast*) indicates forecasted metric anomalies. For more information, see Metric anomaly forecasts.

Tooltip: Hover the cursor over any part of a time-series graph to view tooltip information that is related to multiple anomalies at a particular point in time.

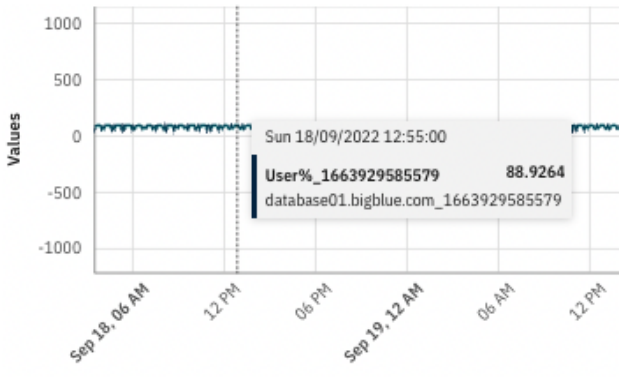


Figure. Tooltip

Chart Icons

Icon	Name	Description
	Copy link	Copies URL link of chart to clipboard.
	Show as table	Tabular representation of anomaly, including occurrence times and values, which can be downloaded as a CSV or JSON file.
	Make fullscreen	Maximizes size of chart to fit onto full page.
	More options	Allows chart to be exported as CSV, PNG or JPG file.

Zooming in on timeline

On the main chart, place and click the "+" "cursor at the beginning of the time period you want to focus on, then drag it along the timeline, to your chosen endpoint. The main chart adjusts to the selected timeline.

For a more controlled zoom, use the zoom scale (miniaturized chart on top of the main chart) to focus on a particular date range. Select and move the vertical slider black bars, left or right:



Figure. Metric anomaly chart zoom

Expanding the timeline of the chart

Change the 2-day timeline in the side panel to a larger, 7-day timeline chart by clicking the **View expanded chart** link that is located under the chart legend.



Figure. Metric anomaly expanded chart with Related alerts and Additional metrics tabs.

Related alerts table

If alerts related to the primary metric are detected, then a tab for *Related alerts* appears in the expanded view of the chart. When this tab is selected, a table of related alerts appears. Up to 4 related alert metrics can be added to the chart as KPIs, in additon to the main metric KPI. For more information, see Related alerts table.

Additional metrics

Use the *Additional metrics* tab to add up to another 3 KPIs to the Metric anomaly details chart.

1. Click **Add metric**.
2. Select an item from each field: *Metric group*, *Metric*, and *Resource*.
3. When selection is complete, the timeline of the KPI is added to the chart.
4. Repeat steps 1-3 up to 2 more times.

To modify a selection, remove the current item by clicking × in the *Metric group*, *Metric*, or *Resource* fields. Then, click inside the empty field and a drop-down displays all available items for that field.

To delete a KPI, click × in the *Metric group* field or the ⊖ icon.

Note: Entering part of a string in a field filters the drop-down menu so that it displays only the items that contain that element. In the following example, in the *Resource* field, entering 02 brings up all resources that contain the string 02.



Figure. Metric search

View selected metrics in their own KPI timeseries chart by clicking the icon, which opens the stand-alone **Metric search** modal. For more information, see Metric search.

Note: If no data is available for a searched metric during the time interval, a KPI is not plotted on the chart.

Related topics

- Metric anomaly forecasts
- Viewing multiple metrics on the metric anomaly chart

Metric anomaly forecasts

Forecasting can indicate when an alert can become critical and help avoid an outage. Use the metric anomaly details chart to view forecasts for a metric anomaly.

Procedure

Open the metric anomaly details chart, as normal (see Viewing metric anomaly details, for more information).

The forecasted trend is the area of the chart that has a dashed (--) line rather than an unbroken (–) line, which indicates historical data. The dashed line might have a purple "confidence interval" around it that shows the likely range of values for the forecast. In some cases, this confidence interval is not shown due to a high level of error that would result in large confidence intervals.

To get more a detailed view of where historical data ends and forecasted data begins, expand the chart by

- clicking and dragging the gray tab with the three vertically aligned dots on the side panel
- or clicking the maximize icon
- or clicking **View expanded chart**

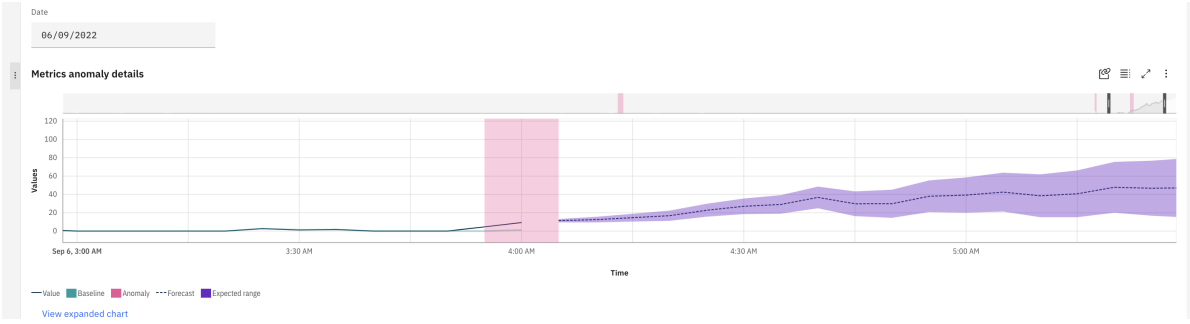


Figure. Expanded and zoomed

When in an enlarged view, you can continue to zoom in on the chart, as normal.

Viewing multiple metrics on the metric anomaly chart

Add multiple metrics (KPIs expressed in a timeseries) to a metric anomaly chart by using the *Related alerts* table or the *Additional metrics* tab. **Note**: The Related alerts table is available when the alert is associated with an Incident.



Figure. Related alerts tab

Procedure

Two methods are available to view the *Related alerts* table. Use the Alert viewer or the Incident viewer.

- To access from Alert viewer
  1. In **Incidents and alerts**, click the **Alerts** tab. Then, open the side panel that contains the metric anomaly details. For more information, see Viewing metric anomaly details.
  2. Click the **View expanded chart** link beneath the chart.
  3. In the expanded view, any alerts (metric and nonmetric) related to the selected alert are listed in the *Related Alerts* table.
- To access from Incident viewer
  1. In **Incidents and alerts**, click the **Incidents** tab.
  2. For the Incident metric you want to investigate, hover over the text in the *Title* column and click the link.
  3. The Incidents *Overview* page opens. Click a metric anomaly in the *Probable cause alerts* or *Top alerts* (when probable cause ranking is not available) column and the metric anomaly chart opens.



Figure. Metric anomaly chart

  4. Click ⧉ to open the chart in its own window.

5. If related alerts are present, they are listed in the *Related alerts* table. Depending on your screen size, you might have to scroll down to view this table.

Related alerts table

Use the checkboxes in this table to enable or disable Key Performance Indicators (KPIs) on the chart.

Note: A KPI is made up of a resource, which is a physical entity such as a network interface card, and a metric. For example, if a server has 2 network interface cards and you want to analyse data for 10 metrics on each network card in the server, the total number of KPIs is 20. KPIs are often interchangeably referred to as metrics, so a metric can sometimes refer to a measurement that is taken for a type of device - for example **`CPU_Context_Switches`**, and sometimes it can refer to an instance of that metric - for example **`CPU_Context_Switches`** on server **`ABC.acme.com`**.

The checkbox of the main KPI is enabled by default and cannot be deselected. The Baseline radio button is also selected by default, but it can be deselected by clicking the radio button (circle icon) of another KPI.

Up to 4 KPIs from the *Related alerts* table can appear on the timeseries chart at the same time.

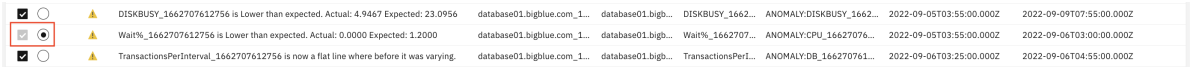


Figure. Multi-Metric chart 2

Note: You cannot apply multiple baselines to a chart. Only one baseline can appear at any one time.

Tooltip: Hover the cursor over any part of a time-series graph to view tooltip information that is related to multiple anomalies at a particular point in time.

First occurrence

When a nonmetric alert is selected on the *Related Alerts* table, a blue vertical line or bar can display, if it is in the same timeline as the metric data displayed. The line indicates the first occurrence of this alert in the selected time period.

Additional metrics

To find and add up to 3 more KPIs to the chart, do the following.

1. Click the *Additional metrics* tab.
2. Find a particular metric by selecting an available value from each field, *Metric group*, *Metric*, and *Resource*.
3. When selection is complete, the KPI is added to the chart.

To modify a selection, remove the current item by clicking ✕ in the *Metric group*, *Metric*, or *Resource* fields. Then, click inside the empty field and a drop-down displays all available items for that field.

To delete a KPI, click ✕ in the *Metric group* field or the ⊖ icon.

Note: Entering part of a string in a field filters the drop-down menu so that it displays only the items that contain that element. In the following example, in the *Resource* field, entering 02 brings up all resources that contain the string 02.



Normalization

The different scales of different metrics can make it difficult to compare them when viewing absolute values together. For example, some metrics can have values of less than 100, while others can have values of more than 10 million. Normalization is used to accommodate comparison, so the scale-bar for the y-axis is not shown. To view absolute values of a metric, uncheck selected metrics until all remaining metric time-series are on the same scale, with the y-axis visible.

Note: If a tooltip is applied to the chart, the absolute value displays, not the normalized value.

Related topics

- Viewing metric anomaly details chart
- Metric anomaly forecasts

Using Metric search to create a timeseries chart

Visualize one or more KPIs, with or without an anomaly, in a Key Performance Indicator (KPI) timeseries chart.

Note: A KPI is made up of a resource, which is a physical entity such as a network interface card, and a metric. For example, if a server has 2 network interface cards and you want to analyse data for 10 metrics on each network card in the server, the total number of KPIs is 20. KPIs are often interchangeably referred to as metrics, so a metric can sometimes refer to a measurement that is taken for a type of device - for example **`CPU_Context_Switches`**, and sometimes it can refer to an instance of that metric - for example **`CPU_Context_Switches`** on server **`ABC.acme.com`**.

Procedure

1. Go to the Automation console.
2. In the **Overview** menu, click **Metric search**.
3. On the **Metric search** page, refine your search to a specific KPI by selecting options from the *Metric group*, *Metric*, and *Resource* menus. The date that is selected represents the endpoint of the chart. KPI.



Figure. Metric search panel

Adding more KPIs to the chart

View up to 5 KPIs on the same timeseries chart.

Assuming that your first-searched metric is plotted on the chart as the primary KPI.

1. Click **Add metric**.
2. Select an item from each field: *Metric group*, *Metric*, and *Resource*.
3. When selection is complete, the KPI timeseries is added to the chart.

4. Repeat steps 1-3 up to 3 more times.

To modify a selection, remove the current item by clicking × in the *Metric group*, *Metric*, or *Resource* fields. Then, click inside the empty field and a drop-down displays all available items for that field.

To delete a KPI, click × in the *Metric group* field or the ⊖ icon.

Note: Entering part of a string in a field filters the drop-down menu so that it displays only the items that contain that element. In the following example, in the *Resource* field, entering 02 brings up all resources that contain the string 02.



Understanding the chart

- The chart displays 7 days of data by default. You can zoom out to see all historical data, including historical data for KPIs that you added to the chart.
- An unbroken black line that is labeled with the name of the KPI that is plotted on the chart represents the values of the selected KPI over time.
- A green-shaded area (labeled *Baseline*) represents the acceptable operating range of that KPI.

Note: The default KPI Baseline is always that of the metric first searched. To change to another Baseline, click the radio button (blank circle) attached to another KPI. Making this Baseline adjustment also changes the KPI that is forecast.

- A red-shaded area or red vertical line (labeled *Anomaly*) indicates where the metric is anomalous: one or more of the anomaly detection algorithms determined the metric was not behaving in accordance with the learned normal behavior.
- A purple area in the chart (labeled *Expected range*), with a dashed line (labeled *Forecast*) indicates forecasted metric anomalies. For more information, see Metric anomaly forecasts.

Related topic

- Viewing metric anomaly details chart
- Metric anomaly forecasts

Displaying analytics details for an alert group

On alert lists associated to an incident, click the option to show the correlation information to understand why these alerts were grouped together. This option displays the underlying temporal, topological, and scope-based groups.

Procedure

1. In the Alert Viewer, click the Click the **View correlation** icon to display the **Correlation** column in the table.
   - **Temporal group column**

     Based on alert history, the alerts in this column that are marked with a large dot ● tend to occur within a short time of each other.
   - **Scope-based group column**

The alerts in this column that are marked with a large dot ● occur within a configurable time window on an administrator defined scope, such as a location, service, or resource.

- **Topological group column**

  The alerts in this column that are marked with a large dot ● occur on resources within a predefined section of your network topology.

2. Click a dot ● to see more details on any of these sub-groups.

Temporal group column

Clicking a dot ● in this column opens the sidebar, with the **Temporal correlation** section open. This section contains the following information:

- **First group instance**

  Date and time of first instance of this group.

- **Total group instances**

  Total number of historical instances of this group. For details of when these instances occurred and how many alerts occurred in each instance, see the Group instance heatmap.

- **Average instance duration**

  Average time in seconds that this group instance lasted.

- **Group instance heatmap**

  Time-based heatmap showing recent historical period in days with a grey square for each day. Each darker square indicates a day on which there was at least one group instance. Hover over the square to see details of this group instance.

Scope-based group column

Clicking a dot ● in this column opens the sidebar, with the **Scope-based** section open. This section contains the following information:

- Group duration

  Duration of the scope-based group. For example: "These alerts were found to share a cause as they all occurred on the same resource within **N** minutes of each other." **N** being the configured time.

Topological group column

Clicking a dot ● in this column opens the sidebar, with the **Topological** section open. This section contains the following information:

- **Topology group name**

  Name of the topology defined in the topology management service, on which this topology group is based.

- **Topology**

  Pane showing the resources in the topology on which this topology group is based. You can perform the following actions on the topology:

  - Click the resource to displays the relationship between that resource and neighboring resources. The relationships are displayed in text on the lines connecting the resources. Examples of relationships include: runsOn, members, exposes.

  - Right-click the resource to display the context menu. For more information about the actions available from the context menu, see see Viewing a topology.

Viewing incidents that are displayed in outgoing connections

View incidents in outgoing connections.

When a new anomaly is detected, IBM Cloud Pak for AIOps generates an incident. If an outgoing connection is configured, the incident is sent to ServiceNow, Slack, or Teams.

- In ServiceNow, an incident is created, and a link to the incident is added to the IBM Cloud Pak for AIOps incident and to the Slack or Teams message if those connections are also configured.
- In Slack or Teams, the IBM Cloud Pak for AIOps incident appears as a message, and a link to the message is added to the IBM Cloud Pak for AIOps incident.

Viewing a ServiceNow incident

1. See the incident section.

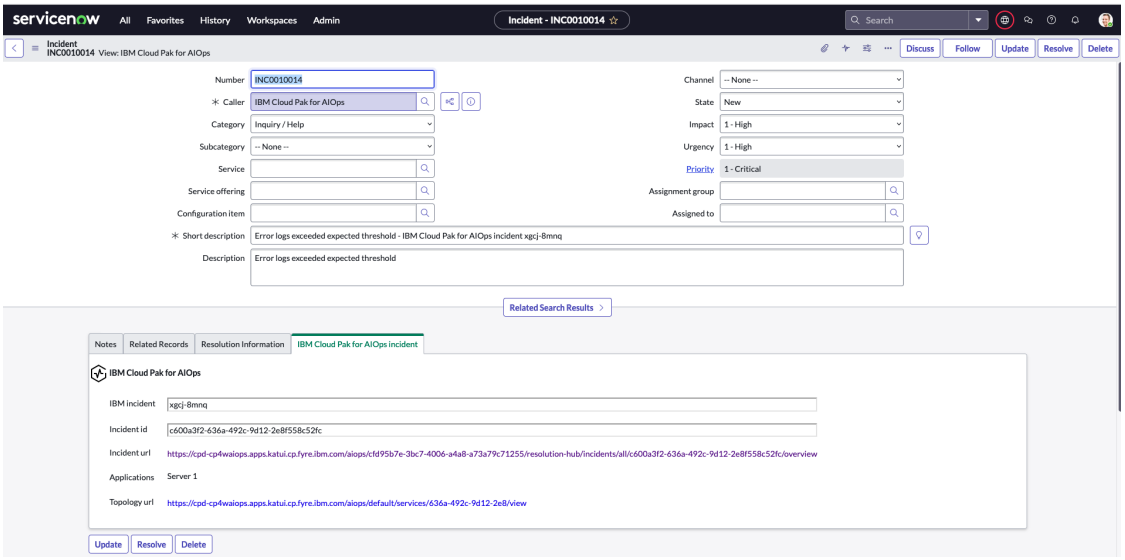


Figure. ServiceNow incident overview

Number: The incident number identifies the incident across IBM Cloud Pak for AIOps, Slack, and ServiceNow.

Caller: The caller is the user who was used to set up the ServiceNow Integration in IBM Cloud Pak for AIOps.

Impact and Urgency: Impact and urgency are set to map the severity of the incident to the priority field of the incident in ServiceNow.

Short description and **Description**: These values are set to the title and long description of the incident. You can change the values in the ServiceNow ticket as well, but the changes are not sent back to IBM Cloud Pak for AIOps.

2. See the IBM Cloud Pak for AIOps section.

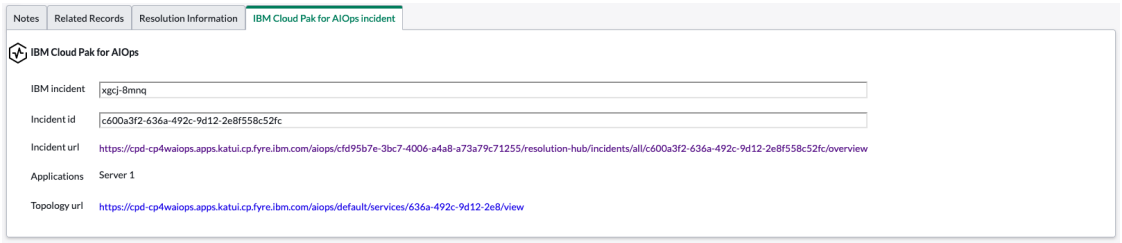


Figure. IBM Cloud Pak for AIOps incident details in ServiceNow

Incident number and ID: The incident number and ID are used to identify the incident in IBM Cloud Pak for AIOps.

Incident URL: The incident URL is a link to the incident in the IBM Cloud Pak for AIOps UI. Click the URL to open the IBM Cloud Pak for AIOps UI to the corresponding incident.

Applications: This value lists the names and URLs of the applications that the incident is associated with. If no applications are detected, **`N/A`** is shown.

Topology URL: This URL links back to the IBM Cloud Pak for AIOps UI to show topology details.

3. See the IBM Cloud Pak for AIOps incidents tab.

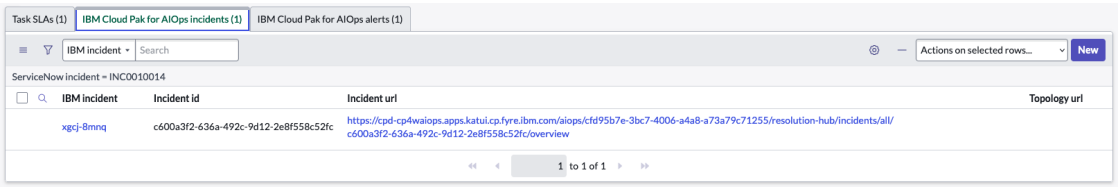


Figure. The IBM Cloud Pak for AIOps incidents tab in ServiceNow

This information is the same as in the IBM Cloud Pak for AIOps section. Click the incident number link to go to the IBM Cloud Pak for AIOps table and view all the IBM Cloud Pak for AIOps incidents in ServiceNow.

4. See the IBM Cloud Pak for AIOps alerts tab.

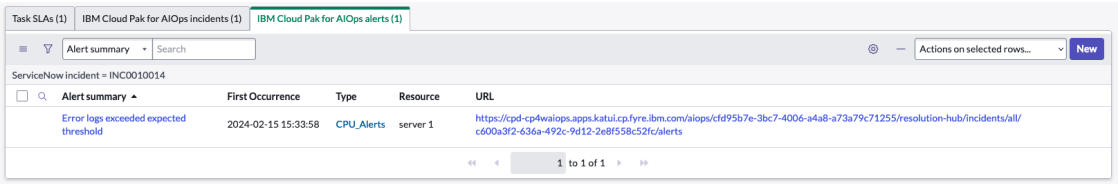


Figure. IBM Cloud Pak for AIOps alerts tab in ServiceNow

IBM Cloud Pak for AIOps incidents are made from a collection of alerts. View alerts that contributed to this incident in the IBM Cloud Pak for AIOps alerts tab.

Event mapping and alert severity

The following table identifies the mappings from IBM Netcool Operations Insight **`alerts.status`** table columns to the IBM Cloud Pak for AIOps event that is sent to lifecycle for processing.

Creating a Netcool connection is a prerequisite to synchronize your event data into IBM Cloud Pak for AIOps. For more information, see Creating Netcool connections.

The second table represents the differences in alert severities between IBM Cloud Pak for AIOps and IBM Netcool Operations Insight.

Event mappings from IBM Netcool Operations Insight to IBM Cloud Pak for AIOps

IBM Cloud Pak for AIOps property	IBM Netcool Operations Insight column	Comment
id	N/A	IBM Cloud Pak for AIOps generated identifier for event.
occurrenceTime	LastOccurrence	
summary	Summary	
severity	Severity	Severity is remapped to IBM Cloud Pak for AIOps scale.
type.classification	[EventId + ':++:' +] AlertGroup	If no EventId is provided, only the value from AlertGroup is used. The separator (:++:) is only present if EventId contributed to the classification value and indicates the extent of EventId and AlertGroup in the string.

IBM Cloud Pak for AIOps property	IBM Netcool Operations Insight column	Comment
type.eventType	Type	
sender.service	Agent	
sender.name	Manager	If the Manager column is not defined, it is set to unknown.
resource.name	Node	If the Node column is not defined, it is the value of NodeAlias column if one has been defined.
resource.hostname	Node	If value is hostname, or NodeAlias if hostname and Node are not defined.
resource.ipaddress	NodeAlias	If the value is an IP address, otherwise not set.
resource.location	Location	
resource.sourceId	BSM_Identity	
resource.service	Service	
resource.port	PhysicalPort	
resource.physicalslot	PhysicalSlot	
resource.physicalcard	PhysicalCard	
resource.interface	LocalPriObj or LocalRootObj	If the value starts with "ifEntry." or "ospfIfEntry.", the trailing interface details will be mapped. Example: "ifEntry.eth0". **Note**: Where values exist for both LocalPriObj and LocalRootObj, the LocalPriObj value will be used, as it is the primary object element of topology. If it has no value, but LocalRootObj is populated, then this will contribute to resource mapping information.
resource.component	LocalPriObj or LocalRootObj	If the value starts with "entPhysicalEntry.", the trailing component details will be mapped. Example: "entPhysicalEntry.name" **Note**: Where values exist for both LocalPriObj and LocalRootObj, the LocalPriObj value will be used, as it is the primary object element of topology. If it has no value, but LocalRootObj is populated, then this will contribute to resource mapping information.
resource.scopeId	ScopeID	
expirySeconds	ExpireTime	
details	alerts.details	The set of details that are stored in details table for event as name-value pairs.

Differences between alert severities in IBM Cloud Pak for AIOps and IBM Netcool Operations Insight

Alert	IBM Netcool Operations Insight severity	IBM Cloud Pak for AIOps severity	Comment
Critical	5	6	
Major	4	5	
Minor	3	4	
Warning	2	3	
Informational	N/A	2	No informational level in IBM Netcool Operations Insight.
Indeterminate	1	1	
Clear	0	N/A	IBM Cloud Pak for AIOps alerts have a state property that indicates an alert is open, clear, or closed. Allows alerts original severity to be maintained.

Managing ChatOps insights

The insights produced by IBM Cloud Pak® for AIOps surface in ChatOps. Whether you use Slack or Microsoft Teams as your ChatOps interface, the experience is consistent with the IBM Cloud Pak for AIOps Platform UI.

For more information about setting up ChatOps connections, see:

- Creating Microsoft Teams connections
- Creating a ServiceNow connection
- Creating Slack connections

Prerequisites

When you configure your ChatOps connection, you must set up two types of channels for your insight flow

- A *reactive* channel for your incidents and alerts
- A *proactive* channel for notifications (from change risk)

Verifying your ChatOps connection

After you create your ChatOps connection, you can use the *reactive* channel to verify that the connection is up and running, by tagging the bot. For example, @**<bot_name>** hello.

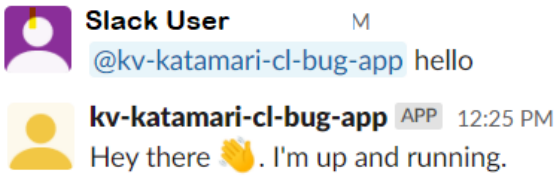


Figure. Slack hello

To get the botName, see Configuring Slack applications or Configuring Microsoft Teams for more details.

In Microsoft Teams, when you enter "hello" the response is "I'm sorry, I don't understand the message". This response indicates that although it does not understand the message, the message was received, so the connection is verified.

Welcome tour

In Slack, after you are added to the *reactive* channel, you will receive a welcome message.

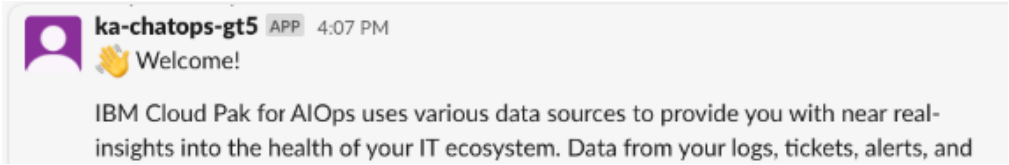


Figure. Slack message

Choose one of the following options:

- *Take tour* Provides an example of incident notification details.
- *Close welcome* Closes the welcome message.
- *Learn more* Links to additional information in the IBM Cloud Pak for AIOps documentation.

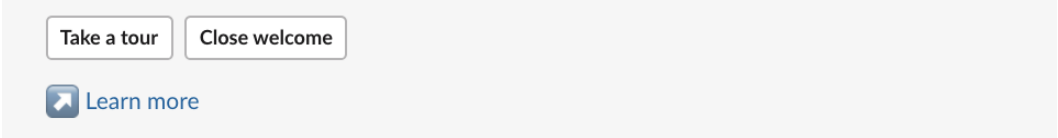


Figure. Slack welcome

In Microsoft Teams, when you receive the welcome notification, select one of the following options to continue:

- *Take tour* Provides an example of incident notification details.
- *Learn more* Links to additional information in the IBM Cloud Pak for AIOps documentation.

Click Teams in the sidebar to access the ChatOps *reactive* channel, anytime.

For more information about working with your insights from your ChatOps connection, see the following topics:

- Managing Incidents In ChatOps
- Probable cause
- Incident Resources
- Recommended runbooks
- Recommended actions
- Viewing alerts in ChatOps
- Change risk in ChatOps

Managing Incidents In ChatOps

Incidents appear in the *reactive* channel of your ChatOps interface. They provide the context for an issue that is severely impacting operations. Included are all alerts that are relevant to the issue, and information about how the affected resources relate to one another. The creation and evolution of incidents is informed by alerts.

Learn more about events, alerts and incidents.

Note: For reasons of convenience and consistency, on this page and on related pages, icons and screen grabs are taken from a Slack implementation of ChatOps.

ChatOps Icons

Icon	Description
	Priority
	Critical Alert (Sev 6)
	Major/Minor Alert (Sev 5 or 4)
	Warning (Sev 3)
	Information/Indeterminate (Sev 2 or 1)
	Link to Platform UI or other online resource
	Probable cause
	Alerts
	Trigger alert
	Recommended runbooks
	Recommended actions
	Incident resources
	Similar past resolution tickets
	Log anomaly

Incident statement

Every incident on a ChatOps interface has an Incident statement area that provides a quick overview of a potential problem. The statement includes a description of the incident, its priority, and when the incident was triggered. The status reflects the current state of the incident and can be set by the SRE.

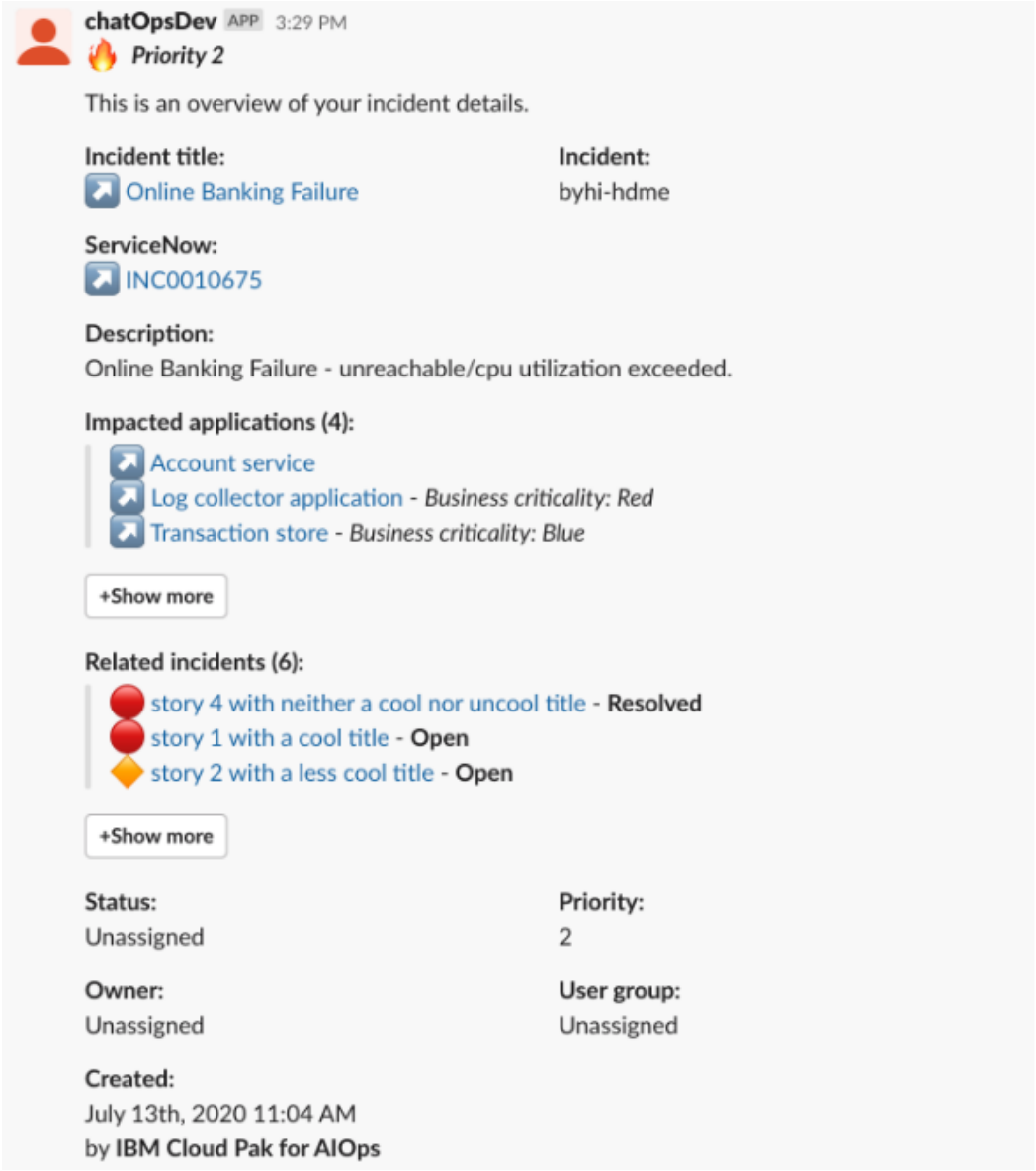


Figure. Incident statement

The Incident statement contains overall information relevant to an incident, including:

Table 1. Incident detection information

Field name	Description
Priority	Priority level that indicated the impact level of an incident on customer.
Incident title	A title that describes the incident. Set as a summary of the first trigger alert, unless it is modified by the user.
Incident	A unique number that is assigned to the incident, also shown within the Platform UI.
ServiceNow	The linked ServiceNow incident number. For more information about configuring ServiceNow connection, see Configuring the ServiceNow App for IBM Cloud Pak for AIOps.
Description	A summary of the incident, including number of alerts and log anomalies.
Impacted applications	List of applications impacted by incident. Access the applications directly that use the hyperlinks.

Field name	Description
Status	*Open*: Incident is open and unassigned *In progress*: Incident is assigned and is being worked on. *Resolved*: Incident is resolved. *Closed*: Incident is closed.
Owner	The email registered to the Slack/Microsoft Teams accounts assigning themselves to the incident. It is recommended that this email matches the email that is registered in the Platform UI.
Created	Date and time that the incident is created.
Updated	Last date and time that the incident is updated.

Updating an incident

At the end of each incident entity, there are options to assign and update the incident.



Figure. Update incident

Click *Edit* to revise incident title, description, group, or priority level. Users can also self-assign the incident and mark it in progress or resolved.

Self-assign and Set to in-progress

The incident owner is set by clicking *Self-Assign* or *Set-in-progress*. When either option is selected, incident ownership is immediately set to the current user. **Note:** It is not currently possible to assign other users as owners.

Tagging users in Slack

When you *Set to in-progress* an incident, you can also tag other users by adding their names to the *Users to notify* field. However, to be tagged, users must be part of the workgroup (**Note:** If even one of those tagged is not in the workgroup, then tagging fails for all users). When tagged, the user is automatically added to the channel and receives notifications to the incident.

Incident resolution and closure

Where an incident status is *Resolved*, it is eventually moved to *Closed* status, in accordance with retention policy period.

Note: When an incident has the status *Resolved* or *Closed*, the functions in ChatOps are limited and no action or link is provided for that incident.

For more information about working with insights from ChatOps, see the following topics:

- Probable Cause
- Incident resources
- Recommended runbooks
- Recommended actions
- Viewing alerts in ChatOps
- Change risk in ChatOps

Probable cause in ChatOps

When an incident is created or updated in response to an issue, IBM Cloud Pak® for AIOps automatically determines which of the related alerts were the likely cause of that issue. For more information, see About probable cause ranking.

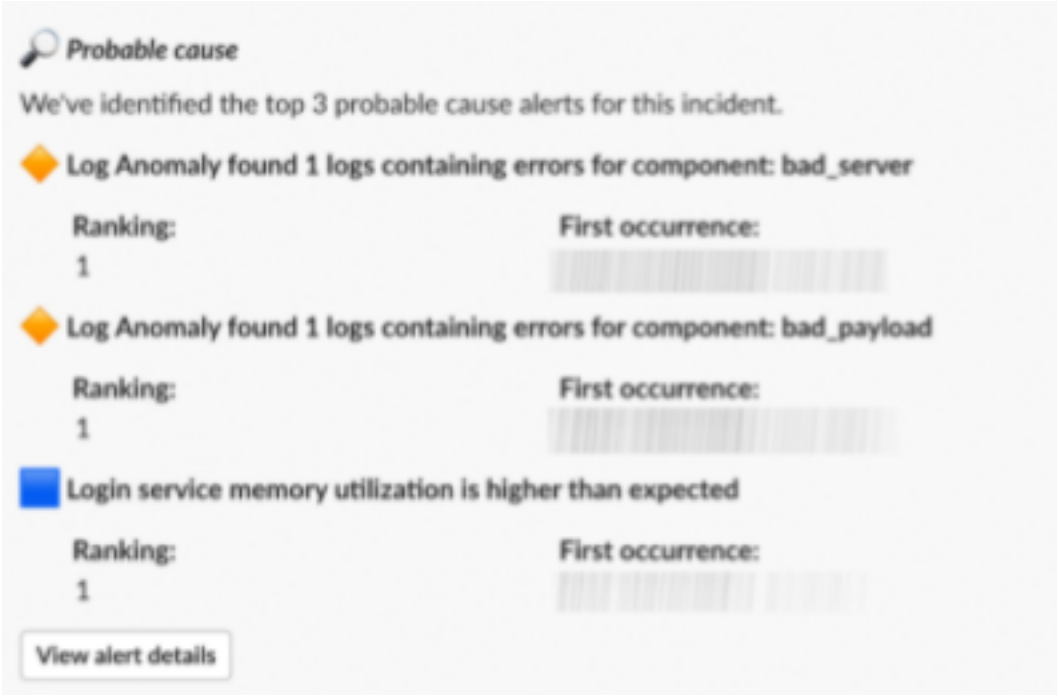


Figure. Probable cause

The *probable cause* section in ChatOps displays the top three alerts, ranked from 1 (highest) to 3 (lowest) in order of probability of being the cause of the incident. To find more information, including severity levels of these alerts, click **View alert details**.

Where *probable cause* alerts cannot be ranked because of similar ranking values, a link is available to the Alert Viewer in the IBM Cloud Pak Automation console, where all probable causes are listed.

Note: Where *probable cause* cannot be identified, the section **Top Alerts** appears instead, listing the top three alerts for the incident ranked by severity and first occurrence. A modal can be opened which shows the expanded alert details of this section.

Next steps

For more information on creating access tokens for the Probable Cause Customization APIs, see Probable Cause Customization API.

Incident resources

The topology view in ChatOps' *Incident resources* provides a holistic view of the relationships between different infrastructure and application components.

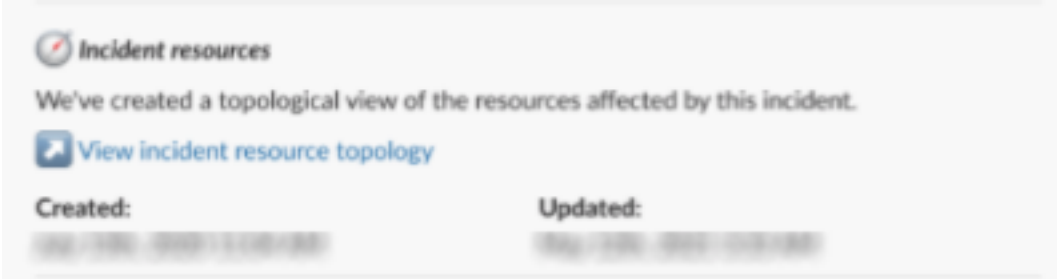


Figure. Incident resources

It can be used to rapidly identify the root cause and the impact of the issue on the wider environment, before any action is taken, avoiding interruption to business-critical activities.

Under *Incident resources*, click **View incident resource topology** to go directly to the Platform UI where you can access topological details about individual entities within an incident.

Note: *Incident resources* is not visible where no application can be associated with the incident.

Associated automations

Runbooks are automated solutions that can be triggered to resolve certain issues. They can be created, edited and updated in IBM Cloud Pak for AIOps.

When an incident surfaces in ChatOps, you can access runbooks associated with this incident. Click the *View Runbook* automations link. This will take you directly to the Platform UI where more information is available about the associated automations:

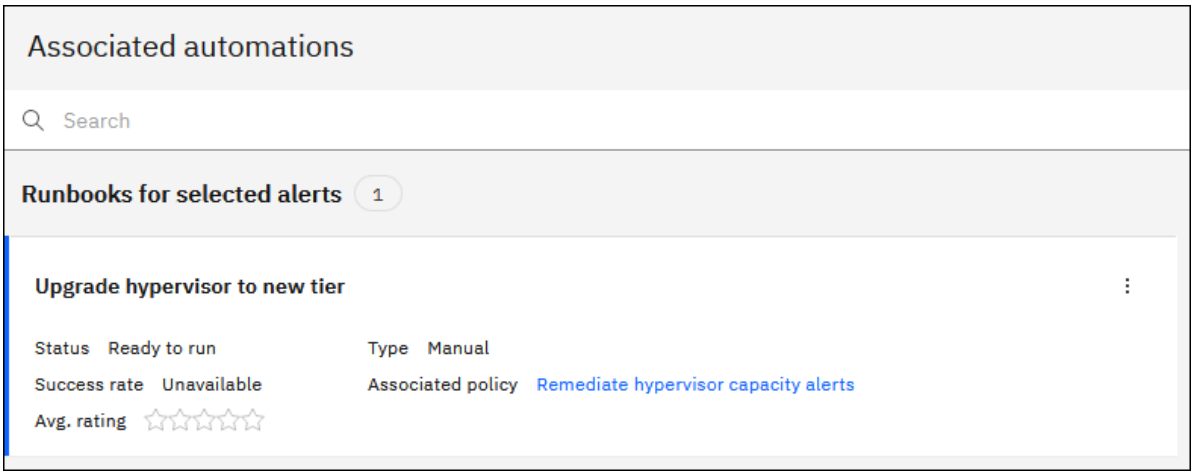


Figure. Associated automations

For more information on runbooks in IBM Cloud Pak for AIOps, see Runbooks.

Recommended actions

ChatOps displays the top three *Recommended actions* based on past resolution tickets that are associated with the incident presented. If you have similar ticket training that is completed for ServiceNow (see Setting up training for similar tickets), you can also search for other past tickets.

Insights and context from your training materials are used to suggest paths to remediation. Use the hyperlinks to go directly to IBM Cloud Pak for AIOps Platform UI.

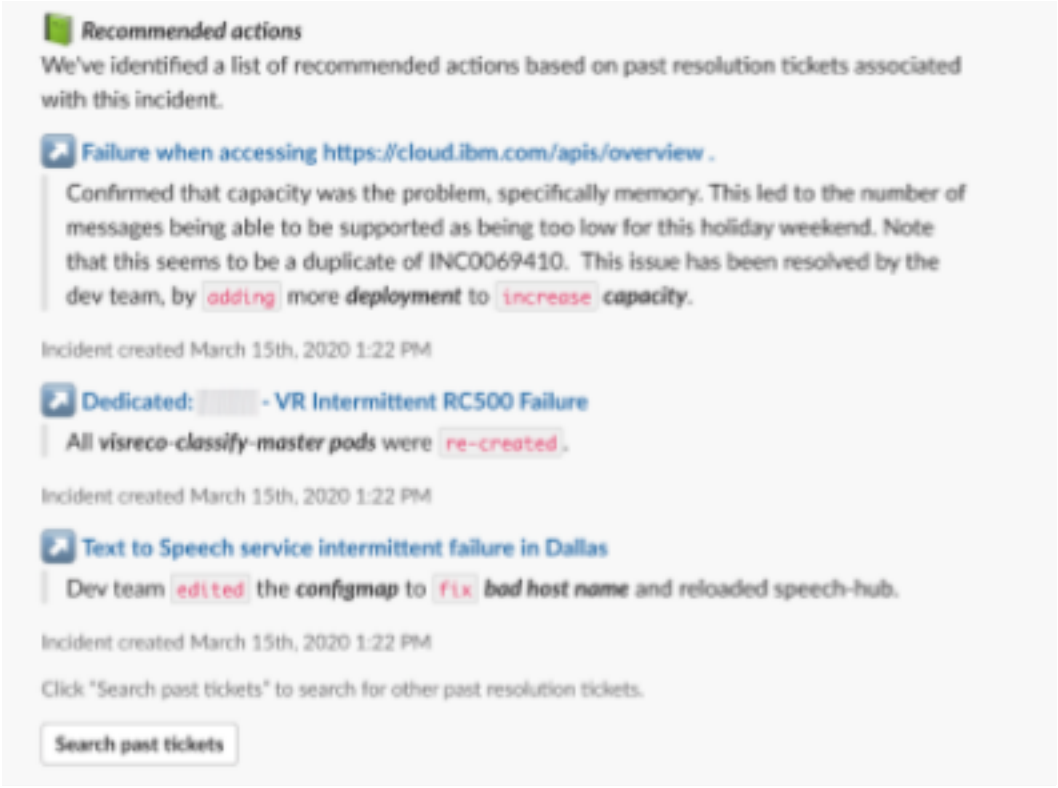


Figure. Recommended actions

Searching recommended actions

Site Reliability Engineers (SRE) can search past tickets in *Recommended actions* to zero-in on likely solutions from similar past tickets.

Search steps

1. Open the *Recommended actions* modal.
2. Click **`Search past tickets`** for any event.
3. Enter search parameters in the *Search past resolution tickets* dialog box.

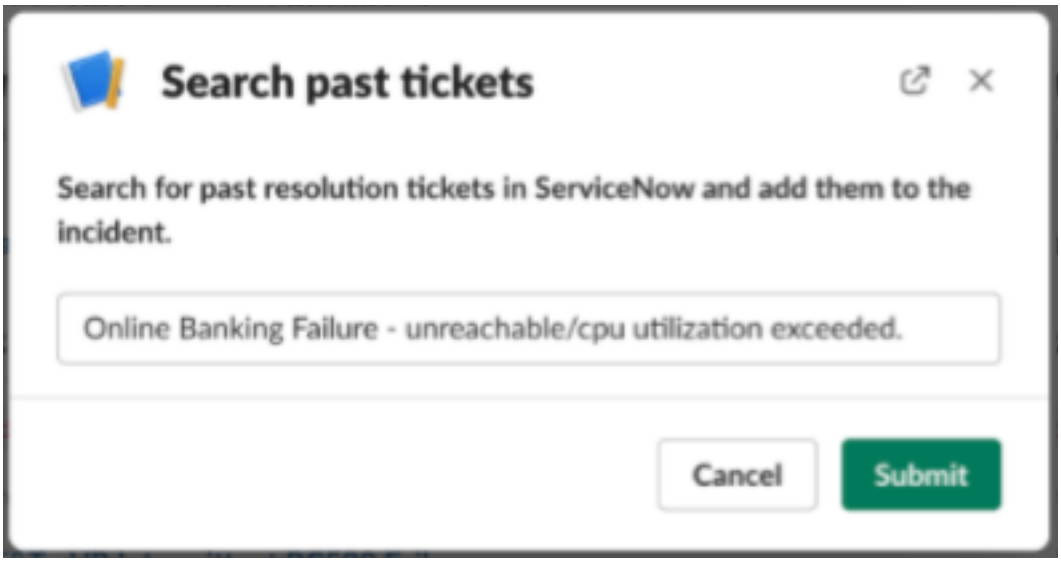


Figure. Search action dialog

Click **Done**. A list of similar tickets appears, with a description for each search result, and details of the action that was taken. You can click the title to go to the issue, or add the issue to the incident by clicking **Add to Incident** .

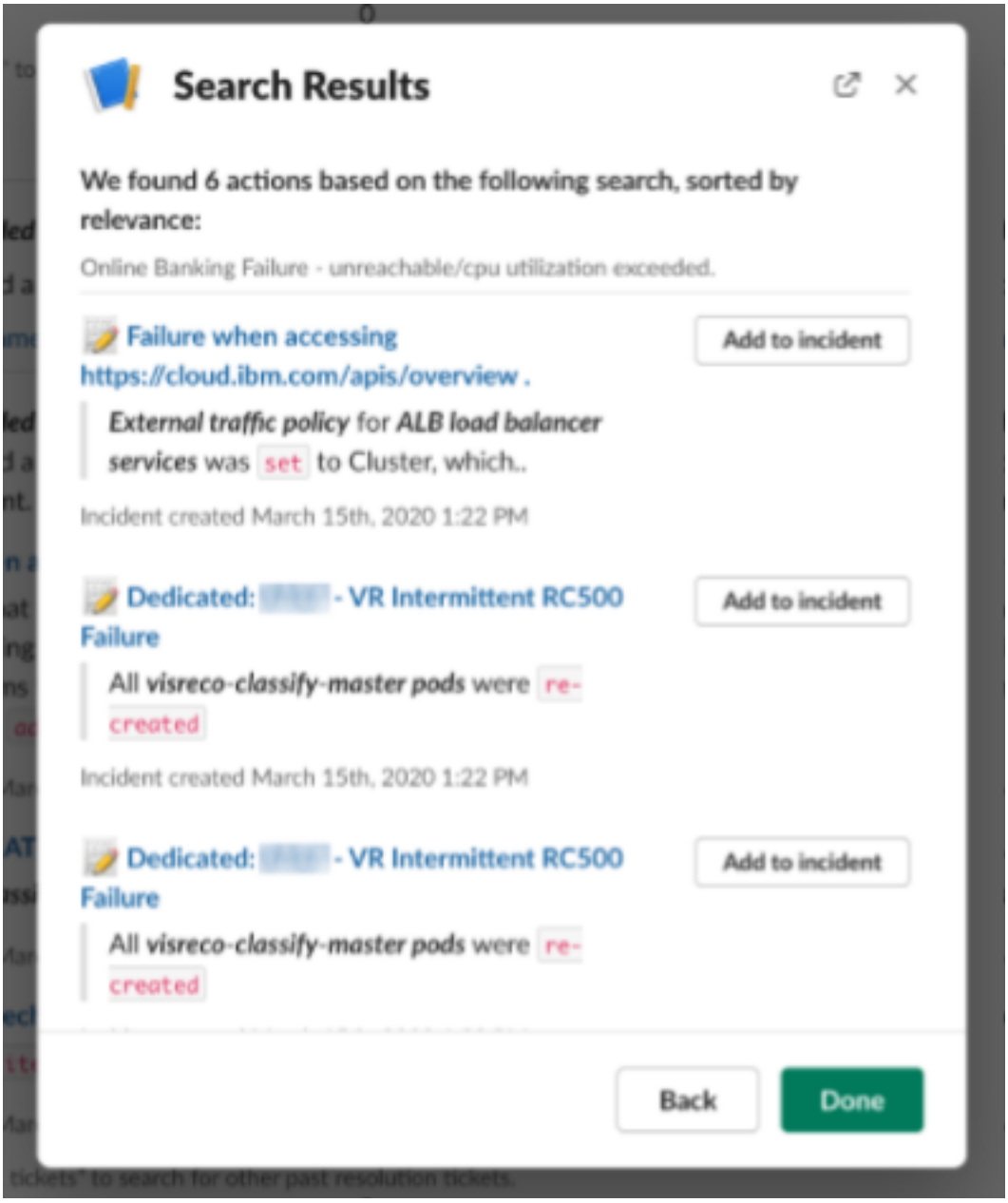


Figure. Search results

Viewing alerts in ChatOps

The *Alerts* area in ChatOps provides details and traceable evidence about alerts. View details of potentially relevant alerts and log anomalies to help you better understand the incident, determine the cause for the alerts, and confirm a diagnosis. Alerts are grouped based on three algorithms: Scope-based, Temporal, and Topological.

- View basic Alerts information
- View detailed information on Alerts
- Investigating log anomalies
- Domain-specific log anomalies

View basic Alerts information

1. Open ChatOps in Slack or Microsoft Teams.

2. Select an incident from your incident (*proactive*) channel, then scroll down to the *Alerts* section of the Incident. This entity presents alerts that are associated with the incident, including their severity level. Click **View in Alert Viewer** in ChatOps to access the *alerts* in IBM Cloud Pak® for AIOps Platform UI.

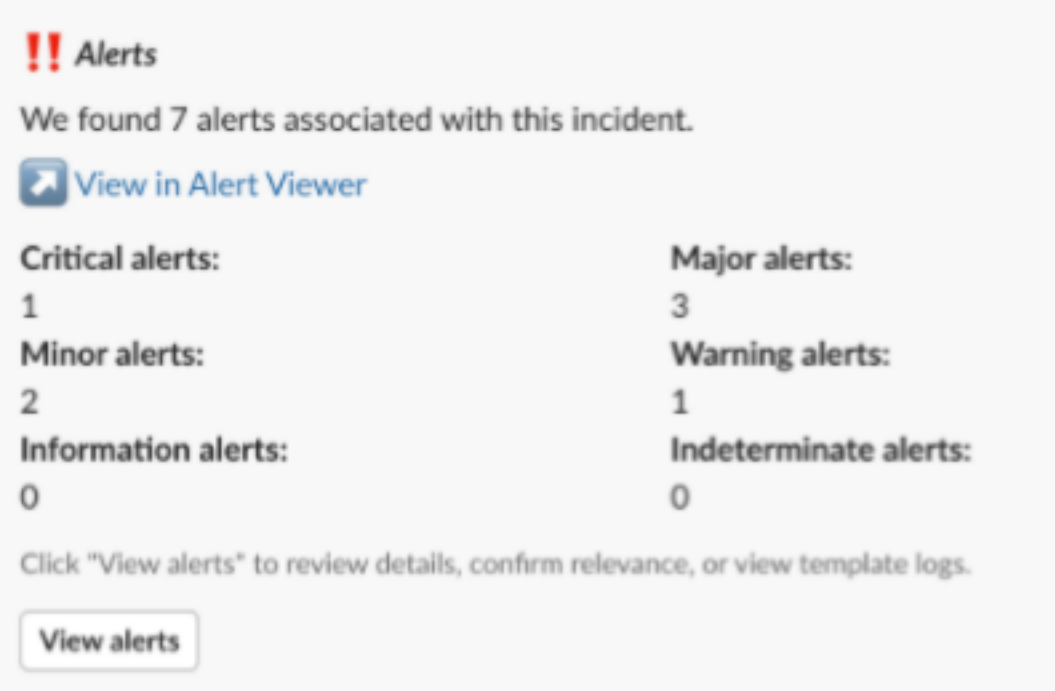


Figure. Alerts

View detailed information on Alerts

In the *Alerts* section of your incident, click **View alerts**. The *View alerts* modal opens.

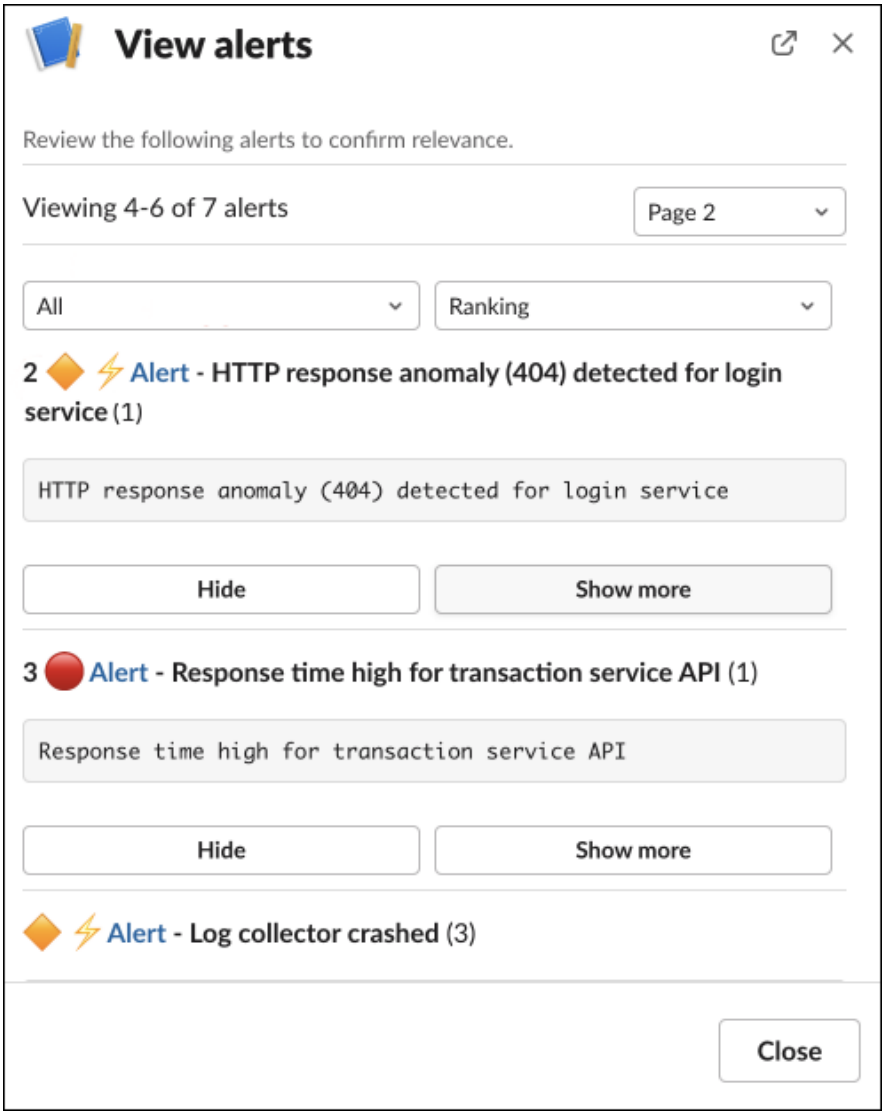


Figure. View alerts

- Use the two dropdown menus to filter alerts by type of alert, severity, status and rank.
- If necessary, scroll down to find all alerts associated with the incident.
- An alert listing can have a number and icon label that is attached to it.

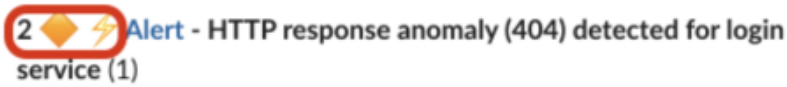


Figure. Icons label

These labels indicate probable cause ranking, severity of alert and if the alert is a trigger alert - indicated by a lightning icon.
Note: These labels are present only if the alert is a trigger alert and/or if it is a probable cause of the incident - or both.

- You can hide any listed alerts for a specific incident entity by clicking **Hide** for that alert. Click **Done** when you are done hiding alerts. **Note**: To see alerts that you hid, select the entity, and click **Unhide**.
- For more information on an alert, click **Show more** on the **View alerts** modal to view the alert's *Status*, *Source*, *First occurrence*, *Last occurrence* and *Resource*, *Ranking* and *Severity*.

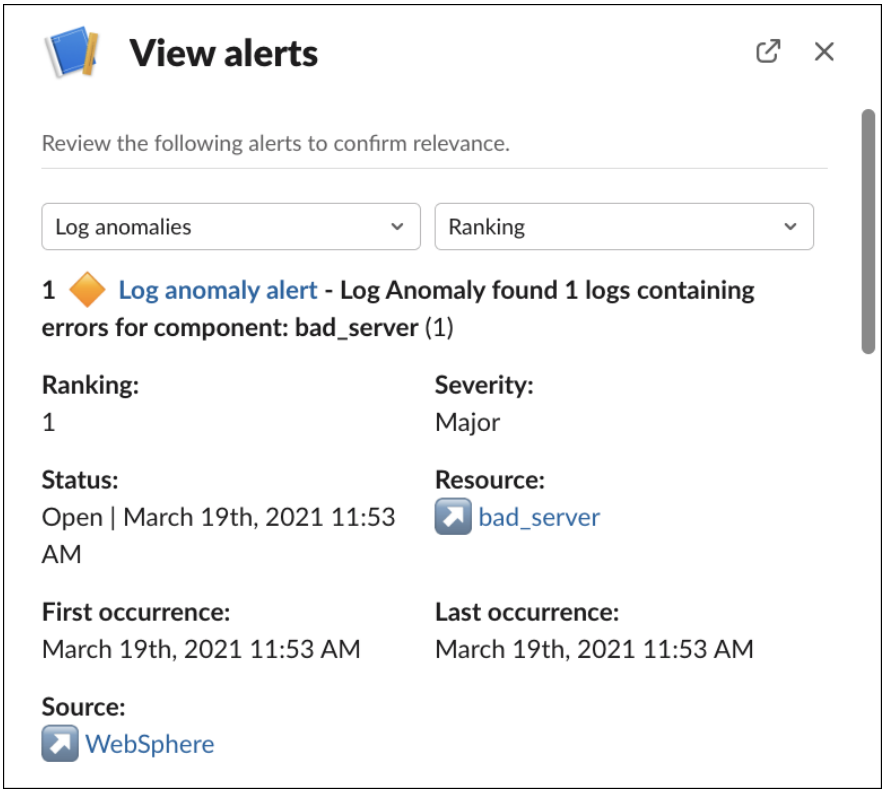


Figure. Show more

Investigating log anomalies

Based on an analysis of your log data performed during AI model training, templates are built that model each type of log message identified. Live log data is analyzed with respect to these templates and any surprises are reported as log anomaly alerts. Here are some examples of log anomaly alerts:

- A log message was expected but is not present in the log data within a time range.
- A log message was not expected but is present in the log data within a time range.
- A log message occurred with a higher or lower frequency than expected within a time range.

Select *Log anomalies* from the dropdown menu in the *View alerts* modal to view the templates that are used to identify anomalous behavior in your log files.

To view the entire log of a log anomaly (the log for all of the templates), click **Attach template logs**. This action attaches the entire log to your Slack channel where you and your collaborators can view the log in its entirety.

This window also reports surprises associated with individual log anomaly templates. For each surprise, the following information is provided.

- Log anomaly **Template or label**. If this log message has never been seen before, one of the following labels will appear:
  - `Unknown_error`: the log message has not been seen before and it contains error information.
  - `Unknown_normal`: the log message has not been seen before but it does not contain error information.
- **Actual**: The number of times this log message actually occurs in the log data within a time range.
- **Expected**: The number of times this log message is expected to occur in the log data within a time range.
- **Preview log**: Click this button to view the actual occurrences in your log. The *Log preview* window shows a maximum of 3000 characters.

  Note: In Microsoft Teams, logs are saved to Microsoft Sharepoint and can be shared from there.

Example 1

Template or label	Actual	Expected

Template or label	Actual	Expected
`stdout F {"name":"@pod75wd5abd11","__in":<NUM>, "hostname":"backend-456b7373c9efgh","pid":<NUM>,"module": "announceCycle/agentHostLookup","level":<NUM>,"msg":"Extremely unusual message for system`	0	3.12

In this example, three log messages matching the template were expected in a given timeframe, but there were no log messages present in the log data. You need to investigate why the expected log messages did not appear. **Note** Expected_count is shown as a float with two digits after the decimal point. This value is calculated based on the statistical analysis described in Workflow of natural language log anomaly detection and Workflow of statistical baseline log anomaly detection.

Example 2

Template or label	Actual	Expected
`Unknown_error`	2	0

In this example, a log message that has never been seen before was present in the live log data, twice within the time range. This unexpected log message also contains error information so should be investigated with urgency.

Domain-specific log anomalies

If an alert is identified as a log anomaly from a specific domain, such as IBM MQ or WebSphere, you can scroll to the (**View Alerts** > **Show more**) modal to find more details about the anomaly. View *Message Code*, *Explanation*, *Categories*, *Frequency of related message codes* and *Recommended actions*, which can contain documentation links that might help resolve the alert.

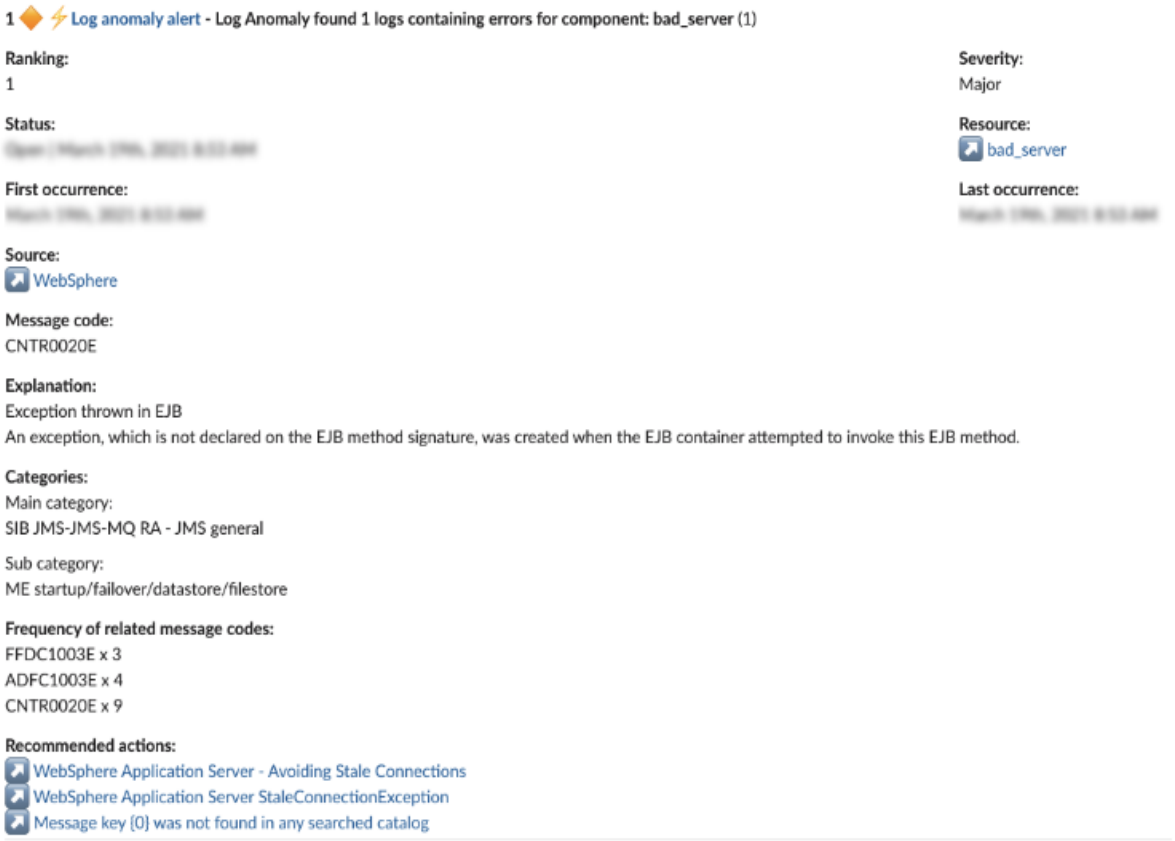


Figure. WebSphere

For more information, see Domain-specific log anomaly alert details.

Change risk in ChatOps

Change risk assessments, which can be notified in ChatOps, provide you with a *Confidence* percentage that represents the confidence level in the AI's prediction of the impact of a change. This percentage is relative to the risk associated with a new or updated ServiceNow change request ticket.

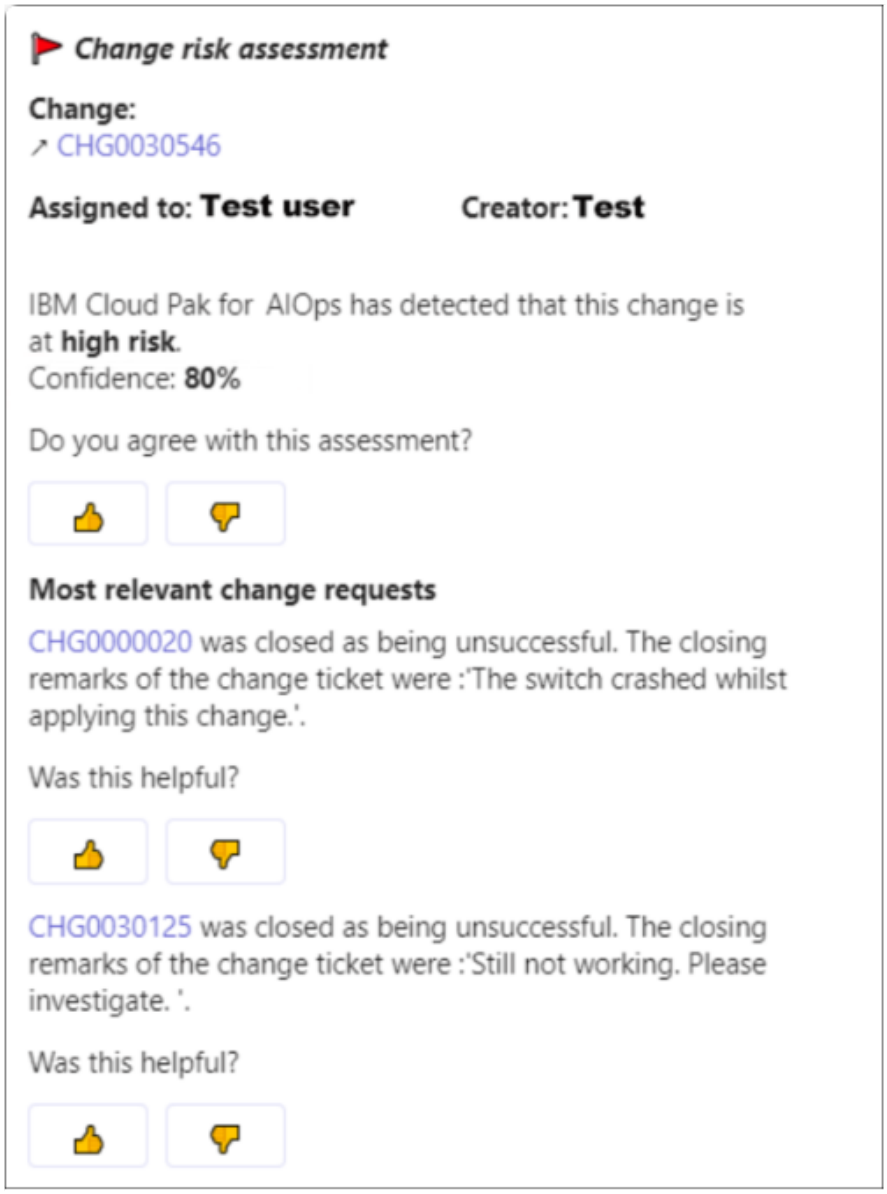


Figure. Chatops change risk

The *Confidence* percentage is calculated based on the change risk AI model and the data that you feed into it. Change can be evaluated based on any type of change, whether it is to code, or to configurations, or to data. If you have historical change request tickets for that type of change, IBM Cloud Pak® for AIOps can provide a confidence percentage for it.

NOTE: Feedback options (thumbs up and thumbs down) appear under the change risk assessment, to allow users to provide input on the assessment and the relevancy of related change requests. Providing feedback helps improve the quality of the change risk model.

Prerequisites

To implement change risk notifications:

- In ServiceNow, any change needs to be recorded in the Change Request Ticket Fields
- In ServiceNow, any deviation from an expected standard of operation must be documented in an incident ticket field

Change Request Ticket Fields

Change request tickets in ServiceNow record detailed information about a change, such as the reason for the change, the priority, the risk, the type of change, and the change category. This information is used to calculate a confidence percentage.

Required fields for change request tickets expect a **`str`** data type value and include the following fields:

- *sys_id*
- *number*
- *assigned_to*
- *sys_created_by*
- *sys_domain*
- *business_service*
- *type*
- *state*
- *short_description*
- *impact*
- *reason*
- *justification*
- *description*
- *backout_plan*
- *close_code*
- *close_notes*

For more information about change request fields, see Change Request in the ServiceNow documentation.

Six incident fields in ServiceNow are scanned for the presence of change ticket numbers: " *description*", "*short_description*", "*work_notes*", "*caused_by*", "*close_notes*", "*comments_and_work_notes*".

Incident Ticket Fields

You create an incident record to document a deviation from an expected standard of operation. The following list describes required fields for incident tickets:

- *number*
- *opened_at*
- *short_description*
- *description*
- *work_notes*
- *caused_by*
- *rfc*
- *close_code*
- *close_notes*
- *closed_at*
- *cmdb_ci*
- *active*
- *assignment_group*
- *comments_and_work_notes*
- *impact*
- *problem_id*
- *priority*
- *severity*
- *state*
- *urgency*

Note: All fields expect an **`str`** data type value, except "*work_notes*", which expects a **`List[str]`** data type.

For more information on incident ticket fields, see Incident Request in the ServiceNow documentation.

Linking Change Tickets and Incidents

Without the correct linkage information generated from these fields, the change risk assessment logic can fail or provide incomplete results. Also, if no linkage is found, the training is marked as unsuccessful or aborted.

When you determine the linkage between the change tickets and the incidents, you also need to look at data interdependence, such as that of the start and end dates. Set the Change Start Date to earlier than the Incident Start Date because the incident occurs after the changes are made.

For more information about change management in ServiceNow, see Change management in the ServiceNow documentation.

In order for you to receive change risk notifications, you must satisfy the following requirements:

1. Set up ServiceNow and your ServiceNow connection
2. Set up your ChatOps connection
3. Train your change risk AI model

Configuring change risk

The more data that you provide, the better the quality of your change risk analysis. At a minimum, your AI modeling requires about 2 percent of change tickets to be problematic. If you train with 20,000–50,000 tickets with a 2–5 percent failure rate, you would yield more robust results. Training does not start automatically when you reach that threshold. If you selected *Live data for initial AI training* in your ServiceNow connection, you would need to manually disable the connection when you reach the minimum threshold of the training model.

The following change ticket fields are used for training. To help produce a good model, provide various values for these change ticket fields:

- *sys_domain*
- *business_service*
- *type*
- *short_description*
- *impact*
- *reason*
- *justification*
- *description*
- *backout_plan*

For more information about training change risk AI models, see Configuring change risk.

Change risk assessment

If all of these steps are complete, when someone opens a change request ticket, a *Change risk assessment* appears in your proactive ChatOps channel. (For more information about channels, see ChatOps insight management). Depending on the risk level, you might also receive a second notification: *Most relevant change requests*.

The *Change risk assessment* includes the ticket number and a link to the change request ticket in ServiceNow and message links to the approver and creator of the ticket.

Confidence percentage

Change risk is scored on a 0–100 percent scale. The higher the percentage, the higher the confidence in the prediction. The amount of information that is provided depends on the severity of the risk.

Most relevant change requests

If the outcome is deemed as high risk change, IBM Cloud Pak® for AIOps provides links to historical relevant problematic tickets. You can view these tickets, with their accompanying explanations, to gain a better understanding as to why the perceived risk is high.

Three types of relevant change tickets can be generated. Each can be identified by a particular term:

1. *successfully completed* – provides details of successfully completed change and incident that is caused by that change. Includes timeline and closing remarks.

2. *closed as being unsuccessful* – occurs when the ticket was not resolved and includes closing remarks, which can indicate that further investigation was undertaken. Also, gives information why this ticket was deemed relevant to new change request.
3. *closed as being successful but change mentioned in incident(s)* – provides details of successfully completed change that was later mentioned in one or more incident tickets.

Managing automations

By creating automations in IBM Cloud Pak® for AIOps, you can proactively set up policies, runbooks, and actions to detect and remediate events, both current and future.

Policies are rules that contain conditions and a set of actions that can be manual or automated. With policies, you can automatically suppress alerts, group alerts based on scope, assign runbooks to alerts, and promote alerts to an incident.

When you create a runbook, you can define the required steps to resolve a situation. If you don't have any runbooks, you can create a new one or load some sample runbooks.

Actions in runbooks are the collection of several manual steps into a single automated entity. An action improves runbook efficiency by automatically performing procedures and operations.

For more information about automations, see the following topics:

- Policies
- Runbooks
- Actions

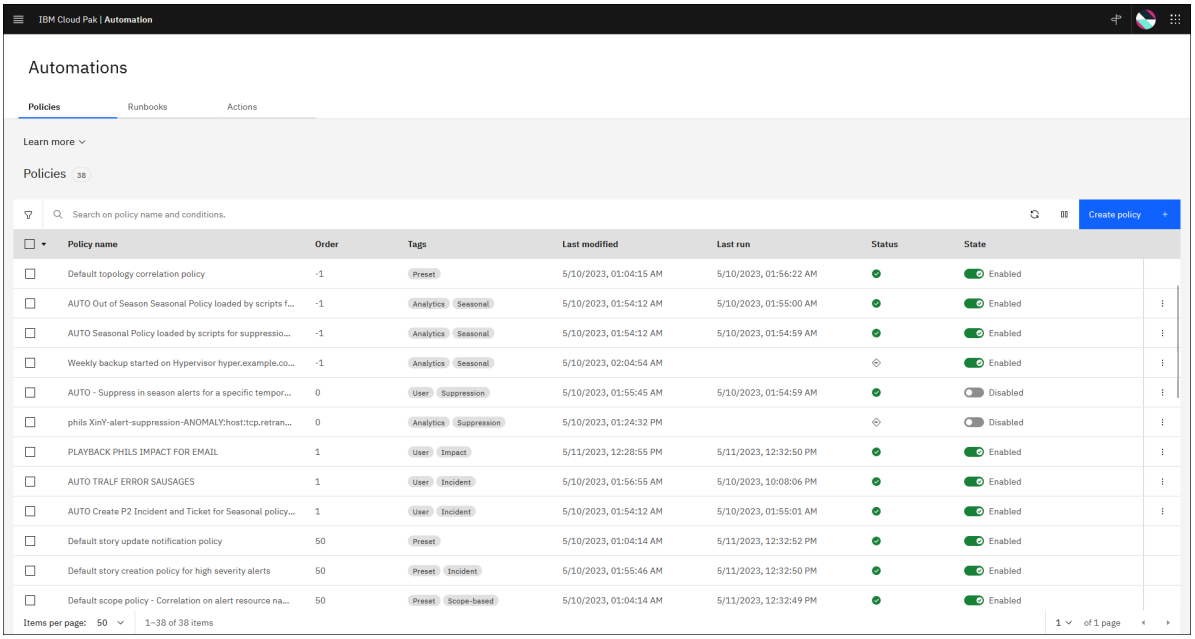
About policies

Policies are rules that contain multiple condition and action sets. They can be triggered to reduce noise by suppressing alerts or grouping alerts together, automatically promote alerts to incidents, assign runbooks to remediate alerts, and invoke IBM Tivoli Netcool/Impact policies to take actions on your alerts.

Policies can be created based on Cloud Pak for AIOps custom alert insights. Using the Policy editor you can build condition sets that are based on Alert properties, Business criticality, and Anomaly insights.

The policies table displays all of the Cloud Pak for AIOps policies that are preset, created by you and your team, or generated by AI analytics. You can enable or disable these policies, create new ones, and edit existing policies. When you click a policy in the table, you can see more details in the side panel. The list of policies is split by pagination, use the navigation arrows at the bottom of the table to move between the next or previous page of policies. To customize the number of rows that are displayed in the table, click the Down chevron icon ⌄ beside **Items per page**.

By default, policies are sorted in the table by **Order** ascending (policies with a lower number are processed first). Where execution order is the same, policies are sorted by **Last run** descending (the most recent policies on top).



Policy UI

Note: Preset policy names and internal action references to "story" or "stories" are synonymous with incidents. Internal references and policy names have no bearing on actual behavior and are for internal processing purposes only. Any references to "story" in preset policy names, details or specifications is not an error, and is equivalent to incident in all respects.

Policies table columns

Policies table columns

Column	Description
Policy name	A name to identify the policy. The default name of a policy is automatically created based on the alert summary field, and specifically on the alert with the highest severity that has ocurred most frequently. To customize a policy name, point the cursor over the name and click Edit. Preset policy names cannot be customized.
Order	Determines the order in which the policy is executed. Execution order is set at policy creation time, on a sliding scale of 1-100. Policies with a lower number are processed first. This is especially important for incident policies. Set the execution order lower for your most important incident policies. Values outside of 1-100 are reserved for suppression, some analytics, and preset policies that handle system needs. **Note:** Actual execution order is a combination of trigger type and execution order. So when the table is sorted by execution order, you might see later **Last run** times on higher priority policies. This is because the processing for different triggers occurs at different times.

Column	Description
Tags	Tags are used to identify the different types of policies. Cloud Pak for AIOps policies are tagged with the following policy types: • Analytics: policies generated by AI analytics. Analytics policies can be viewed, enabled, disabled, and deleted from the Policies table. • Impact: a Cloud Pak for AIOps policy that invokes a IBM Tivoli Netcool/Impact policy, via the IBM Tivoli Netcool/Impact connector. By default, IBM Tivoli Netcool/Impact policies are invoked without input or transformation. Editing options are available in the IBM Tivoli Netcool/Impact policy template. • Incident: policies to create actionable incidents from alerts to improve insights into your issue. • Preset: preset policies define the core behavior of event and alert processing. Preset policies can be viewed, enabled, or disabled from the Policies table. • Runbooks: policies that assign a runbook to alerts for easier automated resolution. • Scope-based: policies that group alerts based on common properties and time of occurrence to identify when issues might be related. • Seasonal: policies that group alerts together that occur within a seasonal time window. • Suppression: suppress alerts that do not need attention to reduce noise and improve team focus. • Temporal: policies that group alerts together that occur within a short time of each other. • User: policies that can be edited by the user. All Runbook and Incident policies, even if tagged as Preset, are also tagged with User. User policies can be viewed, enabled, disabled, modified, and deleted from the Policies table.
Last modified	Displays a timestamp of the last modification to the policy. The date is expressed as month, day, year. The time is expressed as hours:, minutes:, seconds. The time also indicates whether AM or PM. For example, 3/9/2022, 4:45:17 PM. Enabling or disabling a policy, renaming or editing a policy, and adding a comment from the **Journal** tab in the side panel will all update the last modified time stamp.
Last run	Specifies the date and time on which a policy was last run. The date is expressed as month, day, year. The time is expressed as hours:, minutes:, seconds. The time also indicates whether AM or PM. For example, 3/9/2022, 4:45:17 PM. **Note:** If an alert meets the incident-creation conditions of multiple policies, only one incident is created. However, all policies that proposed an incident, and the system incident creation policy will have the same **Last run** time.
Status	The icons in this column indicate one of the following execution states for a policy: • Policy successful, no recent failures • Policy has not been attempted • Policy successful, but has had recent failures • Policy failed on last attempt
State	Denotes if a policy is Enabled or Disabled. You can click the toggle button to change a policy state. Disabled policies do not take any action against incoming alerts. **Note:** Disabling preset policies will prevent some event and alert processing features from working.

Note: A yellow execution status will remain for 8 hours in the event where a policy fails, but then works. Irrespective of how many times the policy works in that 8-hour period. If a policy is "flapping", that is alternating between failing and working, the execution status flips between a yellow and red status.

For more information about policies, see the following topics:

- Searching and filtering policies
- Policy details
- Default incident creation policies
- Default Netcool scope-based grouping
- Creating policies
- Policy editor
- Refreshing the policies table
- Managing multiple policies
- Renaming policies

The following policy templates are available:

- Suppress alerts
- Group alerts based on scope
- Assign a runbook to alerts
- Promote alerts to an incident
- Invoke IBM Tivoli Netcool/Impact

Searching and filtering policies

Text entered in the search field searches for the characters in the policy names and the policy conditions. Note, some conditions are internal, so the search might return policies that you are not expecting. You can also filter the list based on the policy type or current state.

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.

To search for a policy in the table:

Click anywhere in the **Search** field and enter a search term, for example, type "**`hostname`**." The system will show in the table all policies that contain "**`hostname`**" in the policy names and the policy conditions.

Note: Searches are not case-sensitive.

To filter the list of policies in the table:

Click the Filter icon to open the filter side panel.

- Select from the following **Tags** to filter by policy type:
  - Analytics
  - Impact
  - Incident
  - Preset
  - Runbooks
  - Scope-based
  - Seasonal
  - Suppression
  - Temporal
  - User
- Filter policies by **State**:
  - Disabled
  - Enabled
- Filter policies by **Triggers**:
  - Before an alert is created
  - After an alert is created
  - After an alert has been updated

  Within a filter type, multiple selections are summed together. For example, if you select both **Runbooks** and **Incident** under the **Tag** filter, you will see the list of policies that contain **either** of these tags.

Between filter types, the AND operator is used. For example, if you select **Incident** under the **Tag** filter and **Enabled** under the **State** filter, you will see the list of policies that contain **both** of these conditions.

3. Click **Apply** to apply your selected filters to the policies table.
4. Click **Clear filters** above the table header row to clear all searches and filters.

Note: Preset policy names and internal action references to "story" or "stories" are synonymous with incidents. Internal references and policy names have no bearing on actual behavior and are for internal processing purposes only. Any references to "story" in preset policy names, details or specifications is not an error, and is equivalent to incident in all respects.

Policy details

When you click a policy of interest in the table on the Policy UI, a side panel containing multiple information sections opens on the right-hand side of the table. Use the up ↑ and down ↓ arrows to navigate through the list of policies. Where permissions allow, you can also Edit and Delete some policies directly from the side panel. The three tabs that are displayed in the side panel are **Details**, **Journal**, and **Specification**.

- **Details**

  This tab displays the complete set of column data for the selected policy, including familiar fields, such as **Status**, **Tags**, **Created by**, **Last modified by**, **Last run**, and **Execution status**. The **Last failure** is the timestamp that the selected policy last failed. **Number of failures** indicates how many times the policy has failed, and **Last failure details** is the error message. Click **Copy** to copy Last failure details to the clipboard.

  The **Matched** number is incremented by 1 if one or more actions is proposed or performed by a policy. However, when errors occur during policy execution, matched might not be incremented. Therefore, you might see failures but the **Matched** count indicates 0. Matched may only be incremented upon successful policy execution.

  Note: In a case where an alert meets the incident-creation conditions of multiple policies, only one incident is created. However, all policies that proposed an incident, and the system incident creation policy will have the same **Last run** times. Each of these policies will also have their **Matched** counts incremented by one.

- **Journal**

  The Journal tab displays any comments added by operators to record manual changes or updates to the policy. Comments are shown in chronological order. The policy creation date is also displayed.

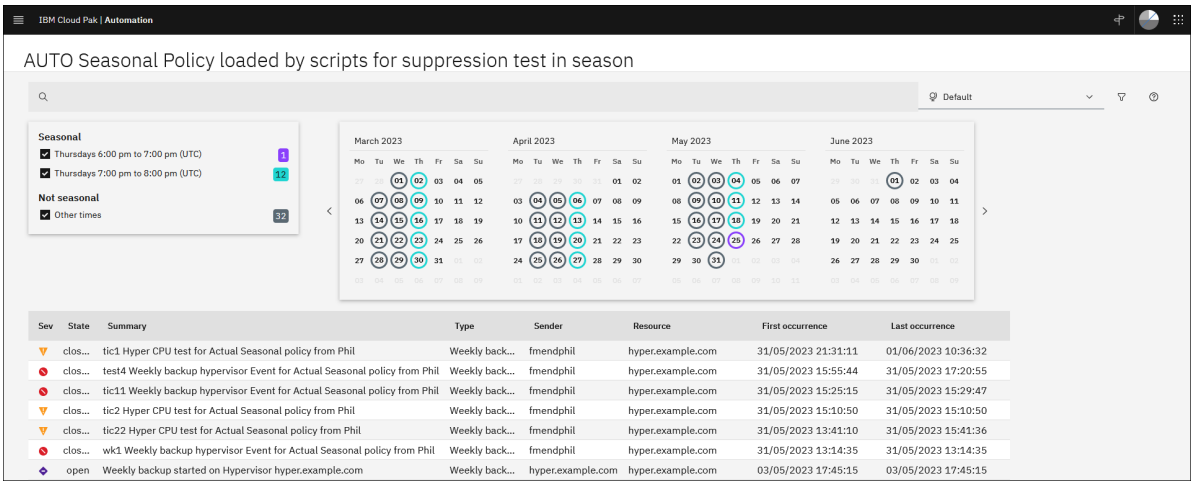
- **Specification**

  Shows you the key information about your conditions and policy. Each condition set of a policy is detailed in the Specification tab, as well as what is created by the policy.

Note: Preset policy names and internal action references to "story" or "stories" are synonymous with incidents. Internal references and policy names have no bearing on actual behavior and are for internal processing purposes only. Any references to "story" in preset policy names, details or specifications is not an error, and is equivalent to incident in all respects.

Seasonality policy details

The seasonality details page provides a calendar view of all of the historical alerts that contributed to the selected seasonal policy. One or more seasonal time windows are listed in the top left pane. Select a seasonal time window to filter the view to show just historical alerts that contributed to that time window.



Seasonality details

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Filter the table to display **Temporal** and/or **Seasonal** policies, or search for a policy by entering a search term in the field provided.
4. Locate the policy of interest and do one of the following to launch out to the policy details page:
   - Click the options menu (three vertical dots) at the end of the policy's table row and then click **View justification**.
   - Click anywhere in the policy's table row and then click **View justification for policy** in the side panel displayed.

The seasonality policy details page contains the following sections:

- **Toolbar**
  - Search: searches alert data in all event group instances that are shown on this page.
  - Filter: filters the alerts that are shown by severity level:
    - Critical
    - Major
    - Minor
    - Warning
    - Informational
    - Indeterminate
- **Seasonal time windows**

  One or more seasonal time windows are listed together with a number indicating how many historical alerts occurred within this time window during the historical period analyzed. The color of the square is the color code for the respective seasonal time window. Select a seasonal time window to filter the Calendar and the Historical alert table to show just historical alerts that contributed to that time window.
- **Calendar**

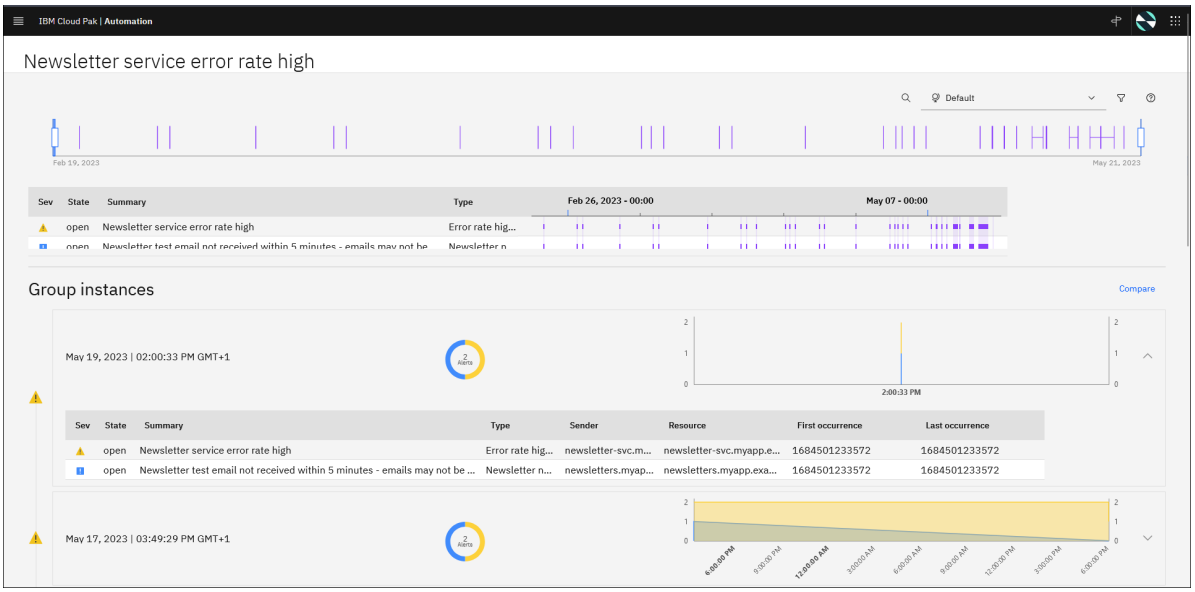
  A calendar of the historical period analyzed (the last three months or more) showing days on which the historical alerts of interest occurred, and colored with the color code for the respective time window. If you hover over any indicated day in the calendar, a tooltip displays a breakdown of the seasonal alerts for that day.

- **Historical event table**

  Table listing the historical alerts during the historical period analyzed.

Temporal policy details

The temporal policy details page shows the alerts that make up a temporal group, together with all of the historical instances of the group. You can navigate into these instances to view detailed timelines of each instance. You can also view alert details.



Temporal policy details

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Filter the table to display **Temporal** and/or **Seasonal** policies, or search for a policy by entering a search term in the field provided.
4. Locate the policy of interest and do one of the following to launch out to the policy details page:
   - Click the options menu (three vertical dots) at the end of the policy's table row and then click **View justification**.
   - Click anywhere in the policy's table row and then click **View justification for policy** in the side panel displayed.

The temporal policy details page contains the following sections:

- **Toolbar**
  - Search: searches alert data in all event group instances that are shown on this page.
  - Filter: filters the alerts that are shown by severity level:
    - Critical
    - Major
    - Minor
    - Warning
    - Informational

    - Indeterminate

- **Overview timeline**

  Displays alert group instances over time and controls the display of alert group instance data on the rest of the page. By default the time range sliders are open sufficiently to show data on all alert group instances. Modify the time range by either clicking and dragging over the desired range inside the timeline, or by dragging the sliders to the desired range. The rest of the screen updates accordingly.

- **Alert group instance timeline**

  Displays all of the alerts that have historically participated in instances of this temporal alert group. The instance map provides a graphical view over time of when the various instances have occurred.

- **Alert group instance details**

  Displays the following information for each alert group instance:

  - Start date and time of event group instance: indicates the first occurrence value of the first alert in the alert group instance.
  - Distribution of event severity values: pie chart providing a visual indication of the alert severity values. Hover over the pie chart for more details.
  - Sparkline: chart of alert occurrence over time.
  - Duration of event group instance: duration of the alert group instance, in text.
  - Down chevron icon: click the down chevron icon to see an alert table showing column details for each alert in this group instance.

Default incident creation policies

IBM Cloud Pak® for AIOps provides two preset (or default) incident creation policies. For incident creation to occur, you must enable one of these default policies, or create your own custom incident creation policy. These policies are disabled by default.

Default incident creation policy for high severity alerts

This policy proposes an incident based on a Major or Critical severity alert. The policy contains one condition set and action.

Condition set 1

If this condition is met:

Property	Operator	Matches	Value
Value of: alert.severity	greater or equal	only	5-Major

The following action is taken: Create a **Priority 1** incident.

Default incident creation policy for all alerts

This policy proposes an incident based on any alert. The policy contains the following condition sets and actions.

Note: In incident policies, only the actions from the first matching condition set are executed.

Condition set 1

If this condition is met:

Property	Operator	Matches	Value
Value of: alert.severity	equal to	only	1-Indeterminate

The following action is taken: Create a **Priority 5** incident.

Condition set 2

If this condition is met:

Property	Operator	Matches	Value
Value of: alert.severity	equal to	only	2-Information

The following action is taken: Create a **Priority 4** incident.

Condition set 3

If this condition is met:

Property	Operator	Matches	Value
Value of: alert.severity	equal to	only	3-Warning

The following action is taken: Create a **Priority 3** incident.

Condition set 4

If this condition is met:

Property	Operator	Matches	Value
Value of: alert.severity	equal to	only	4-Minor

The following action is taken: Create a **Priority 2** incident.

Condition set 5

If this condition is met:

Property	Operator	Matches	Value
Value of: alert.severity	greater or equal	only	5-Major

The following action is taken: Create a **Priority 1** incident.

Incident Creation based on Persistence and Golden Signal

BETA *Incident creation based on persistence and golden signal* is available only for technology preview, not for production usage.

Incident prioritization uses the golden signal policy and the alert suppression policy to filter out incidents when the observed alerts contain a golden signal and are not suppressed. The policy suppresses alerts that do not need attention.

An incident is only created when the following criteria are met:

1. Events persist for a significant duration, leading to an unsuppressed alert.
2. The golden signal that is associated with the alert is of the type `EFFECT`.

Prerequisites

Configuration prerequisites

1. Administrators can enable X in Y. Create a feature flag configmap that contains the flag or update an existing configmap to contain the `IBM_IR_AI_X_IN_Y_PREVIEW_ENABLED` term.

```
apiVersion: v1
data:
  IBM_IR_AI_X_IN_Y_PREVIEW_ENABLED: "true"
kind: ConfigMap
metadata:
  name: feature-flag-configmap
```

2. Restart the **`ibm-ir-ai-operator-controller-manager`** pod if you are not running these steps before installation or upgrading.

Data prerequisites

Historical alert data is used in the training process to learn values of X and Y, which are later employed for creating XinY alert suppression policies.

1. In the Automation console, click **AI model management** and find the *Alert suppression XinY policies* tile. If this tile does not appear, restart the **`aimanager-aio-ai-platform-api-server`** pod. For more information, see The Alert suppression XinY policies tile does not appear in AI model management.
2. Click the tile and follow the steps to train the model and then deploy it. For more information, see training a model.

This instantiates the policy and adds it to the policy list in **Automations**.

Note: To create effective Alert Suppression policies, a minimum of 7 days' worth of historical alert data is needed. This data can be supplied either during the training phase or continuously gathered from the alerts that are received after integrating your application with AIOps.

About this task

After you train your model with historical data, you are ready to proceed to using the incident creation based on persistence and golden signal policy.

Procedure

1. Go to the **Automations** page and click the **Policy** tab to see all available policies.
2. Click the *Tag* column and from the dropdown list check *Suppression*.
3. Any policies that have suppressions are listed. Find the policy that contains the prefix *X-in-Y-alert-suppression*. If it is not there, try clicking the refresh icon, to update the policy list.
4. Click the *X-in-Y-alert-suppression* policy to open a sidebar information panel. In the panel, slide the **Status** switch to **Enabled**.

   Note: The description in the panel gives the X (number of events) and Y (timeline) values that are deduced from training the model. You can edit these values. For more information, see Edit Policy.
5. Next, close the panel, click the *Tag* header column, clear *Suppression*, and check the *Incident* box.
6. Among the incident policy listings are the *Golden Signal and Persistence incident creation* policy. Click this policy to open the sidebar panel.
7. In the sidebar panel, set the **Status** switch to **Enabled**.
8. The *Golden Signal and Persistence incident creation* must be the only *Incident* tagged policy that is enabled. If other *Incident* policies are enabled, then click them, in turn, and disable them in the sidebar panel.
9. You might need to wait before an anomaly is generated as an alert. Alerts that are generated from a golden signals policy might be delayed for a few minutes.

You now have a policy that combines the two policy values when creating incidents. Now when alerts come in, they must fulfill the conditions of the two policies to be promoted as incidents:

- They need to contain a Golden Signal **`Error`**
- The X and Y values must be outside the allowed range set when the model was trained – for example, where the model has the value of 12 events within 60 minutes, an anomaly only surfaces when the events are 12 or above.

Creating policies

Click the **Create policy** button and choose from a set of templates to help you get started.

See the following topics for more information about creating policies:

- Suppress alerts
- Group alerts based on scope
- Assign a runbook to alerts
- Promote alerts to an incident
- Invoke an IBM Tivoli Netcool/Impact policy

For a detailed breakdown of the policy editor, see Editing policies.

Suppress alerts

Some alerts might not be important for monitoring your organization's environment. For alerts that do not need to be viewed or acted on, use suppression policies to reduce noise for your operations teams by suppressing unwanted alerts. When an alert is suppressed in Cloud Pak for AIOps, the **`alert.suppressed`** flag is set to true. The alert is still present in the system and can be viewed in the Alert Viewer, but the alert is filtered out of the view by default. Incident creation and runbook policies do not run for suppressed alerts. However, scope-based grouping policies continue to run so there is a scenario where suppressed alerts can be included in incidents.

You can also give feedback to Cloud Pak for AIOps to indicate whether an alert was correctly identified and add a suppression policy to suppress low voted items. For more information about giving feedback, see Alert details.

Enabling X in Y alert rate suppression

Many alerts can represent problems where performance thresholds were breached, for example, a switch that reaches its maximum bandwidth capacity during a peak period. These temporary threshold breaches are often expected, and the business policy might be to ignore them unless the breach is sustained for a prolonged but specific period. However, if such alerts are sustained beyond these specified time periods, operators must take action. In this classic X in Y alert rate scenario, the operator is only interested in alerts that occur more than X times in Y minutes. When the alert rate breaches the threshold, the described alerts should not be suppressed. Until the threshold is breached, the alerts do not need to be acted upon.

In suppression policies, you enable X in Y alert rate suppression under actions in the policy editor. Select **Suppress the described alerts until:**. Then, specify your values for X (number of alerts) and Y (minutes).

Attention: Please be aware about alert states that contribute to the X count. The X value is incremented for deduplicated problem and resolution alerts. Therefore, if a deduplicated alert flaps 10 times (alternating between open and clear) "X" would contain a value of 20 (not 10 for just the problem alerts).

Limitations

While the suppression policy can be applied to all alerts, and all alerts can use the Alert property condition to identify when to suppress an alert, use of the Alert insight conditions have certain limitations.

- Anomaly insights are available on all metric anomaly alerts and on certain log anomaly alerts.
- Only a subset of log anomaly alerts identify an exception or error code.
- The statistical baseline log anomaly algorithm typically identifies these exceptions or error codes.

About this task

In this example, preproduction, development, and test environments are monitored by the same monitoring tools as the production environments. The development and test environments naturally have ongoing changes that might trigger various warnings and notifications all the time. These warnings and notifications are relayed to Cloud Pak for AIOps as alerts. However, these alerts do not require action as they are from resources that change constantly as part of the preproduction work. The monitoring tools are usually set up to avoid sending such alerts to the operations teams. However, if such alert information is sent to Cloud Pak for AIOps, for example, due to a misconfiguration, then the operations team might face unnecessary noise.

To save your operations team from being distracted by alerts from such environments, you can create a suppression policy to suppress alerts that come from the monitored resources that make up these environments.

Important:

- Unlike other Cloud Pak for AIOps policy templates, suppression policies contain a single condition set.
- The supported **Anomaly insights** properties for suppression policies are displayed in the tree view of the **Property** field. To display the available options, click **Add condition** > **Alert insight** and place the cursor in the Property field. For more information, see Supported Anomaly insights properties.

Example

1. Click the navigation icon to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Create policy**.
4. Click the **Suppress alerts** tile.
5. Enter a name in **Policy name**, for example, the name can be "Suppress events from preproduction environment". You can also add an explanation of the policy in **Description** to help you and others understand the purpose of the policy, for example, "Suppress alerts from hosts in the preproduction environment to prevent incidents from being created for development and test hosts".

   User-created suppression policies are always the first user policies to be run. The order is set to 0 and cannot be changed.
6. To define the conditions for alerts to activate alert suppression, click **Add condition** and select **Alert property** or **Alert insight** attributes. For this example, Alert property is used.
7. For each attribute, select values for **Property**, **Operator**, **Matches**, and **Value**. For **Alert insights** attributes, also select an **Insight type** value, such as **Anomaly** or **Business criticality**.

   For this example, select **`alert.resource.hostname`** from the **Property** list .You can type **`hostname`**, and the system shows in the property list all alert properties that contain the text **`hostname`**. Then, select **`alert.resource.hostname`**.
8. From the **Operator** list, select **starts with**.
9. From the **Matches** list, select **any of**.
10. In the **Value** field, enter **`dev`** and select **`String: dev`**. In the same field, enter **`test`** and select **`String: test`**.

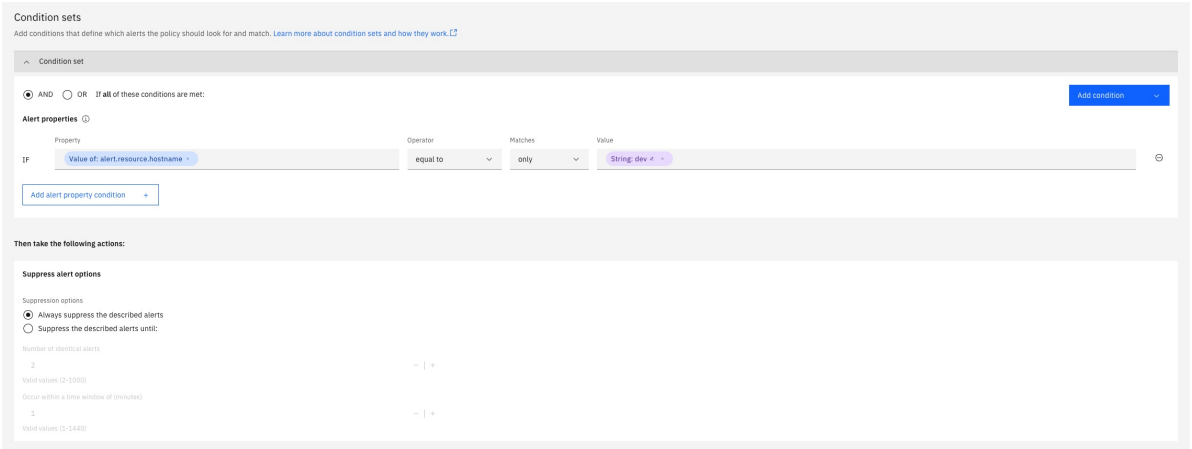


Figure. Supression policy

11. Select **Always suppress the described alerts** or **Suppress the described alerts until** and specify the number of identical alerts or how many alerts occur within a time window of minutes.
12. Click **Create policy**.

New and updated policies can take up to 1 - 3 minutes to take effect.

Note: Suppression occurs on the alert level, not the event level. If a suppression policy for a specific condition is enabled, any matching alerts are suppressed. If the policy is removed later, any future matching alerts are not suppressed, and any existing alerts are not unsuppressed. If the suppressed alert is not closed, for example, if the problem is still active, any future events are deduplicated into the suppressed alert until the problem is resolved.

Scenario where alerts can be both suppressed and part of an incident

Suppressed alerts are not typically included in incidents. However, take a scenario where a group of alerts are subject to a scope-based policy. One alert in the group is not suppressed, and also matches the conditions of an incident policy. Then it is possible for the other alerts to show suppressed and in an incident. They are added to the incident because the scope grouping makes them contextual alerts for that incident.

In this image the one alert which is not suppressed is subject to the same scope-based policy as all the suppressed alerts. If that alert also matches an incident policy, then the other "suppressed" alerts are all included in the incident. This is expected as scope policies still run for suppressed alerts. On the Alert Viewer then it is possible if these circumstances occur for alerts to be both suppressed and part of an incident.

In incident	Suppressed
Yes	Yes
Yes	Yes
Yes	No
Yes	Yes
Yes	Yes
Yes	Yes

Suppressed alerts and part of an incident

Suppress seasonal alerts

Generate a new suppression policy to suppress the in-season alerts identified by an analytics-created seaonal policy. After the suppression policy is created, you can edit it to add conditions if needed.

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**. The Policies table is displayed.
3. Click the Filter icon to open the filter side panel.
4. Select the **Seasonal** tag and click **Apply**.
5. In the Policies table, click anywhere in the row of the analytics-created seaonal policy that you want to generate a suppression policy for.
6. In the side panel displayed, click the plus icon "**+**" beside **Generate suppression policy**.
7. Enter a name for your policy that is 1-120 characters. Leading and trailing blank spaces will be removed from the policy name if provided.
8. Click **Generate policy**.

The new suppression policy is created and available in the Policies table. You can edit the policy to add conditions if required.

To see how many suppression policies or other policies reference a seasonal policy ID, complete the following steps:

1. In the Policies table, select the seasonal policy of interest.
2. If one or more related policies are found, click the search icon for **All related policies**.

The list displayed is generated by searching all policy names and conditions for the policy ID of the selected seasonal policy. You can click **Clear filters and show policies in table** to launch out to the related policies in the Policies table.

Suppress seasonal alerts using the policy templates

You can also create your own suppression policy for seasonal alerts by using the policy template and condition builder.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Create policy**.
4. Click the **Suppress alerts** tile.
5. Enter a **Policy name** and optionally a **Description**.
6. Suppress seasonal alerts by making the following selections in a condition set:
    1. Click **Add condition** and select **Alert insight**.
    2. Set the Insight type to **Seasonal**.
    3. From the Property drop-down list, select `insight.details.policyId`.
    4. Set the Operator to **equal to**.
    5. Set Matches to **only**.
    6. Set the Value to the seasonal policy ID.

For more information about alert seasonality, see Displaying alert seasonality.

Group alerts based on scope

Group alerts based on common properties and time of occurrence to identify when issues might be related. When related alerts are grouped based on their scope, you can view the details in the Alert Viewer.

The default scope-based policy in Cloud Pak for AIOps is a correlation on alert.resource.name constrained by a rolling window of 15 minutes. By using this template, you can override the default scope-based groupings.

About this task

In this example, you want to scope alerts by their cluster name and by their location.

Example

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Create policy**.
4. Click the **Group alerts based on scope** tile.
5. Enter a name in **Policy name**, for example, "Sales region 2 scope policy". You can also add an explanation of the policy in **Description** to help you and others understand the purpose of the policy, for example, "Scope alerts from the sales region 2 cluster by location".
6. Set the **Execution order** to higher.
7. Define the following conditions for alerts that will activate scope-based alert grouping, based on the ScopeID that is specified:
    1. Click **Add condition** and select **Alert property**.

2. From the Property drop-down list, select **alert.resource.cluster**. You can type "cluster" and the system will show in the property drop-down list all alert properties that contain the text "cluster", which in this case is two properties.
3. From the Operator drop-down list, select **equal to**. From the Matches drop-down list, select **only**.
4. In the Values field, enter **SALES.REGION2.CLUSTER** and select **String:SALES.REGION2.CLUSTER**.

8. Add alert properties, strings, or both to create the ScopeID for this group. These properties define the scope for how the alerts are grouped. In this example, the two alert fields in **ScopeID** are separated with string ":". This will group all alerts from the cluster (SALES.REGION2.CLUSTER), but sub-divide them based on their locations (for example, Dallas, New York).

   For this example specify the following ScopeID:

   1. Click anywhere in the properties field and select **alert.resource.cluster** (or type "cluster" and select it from the options displayed).
   2. In the same field, enter a colon **":"** and select **String::**.
   3. Finally, click in the field again and select **alert.resource.location** (or type "location" and select it from the options displayed).

   Example scope-based policy

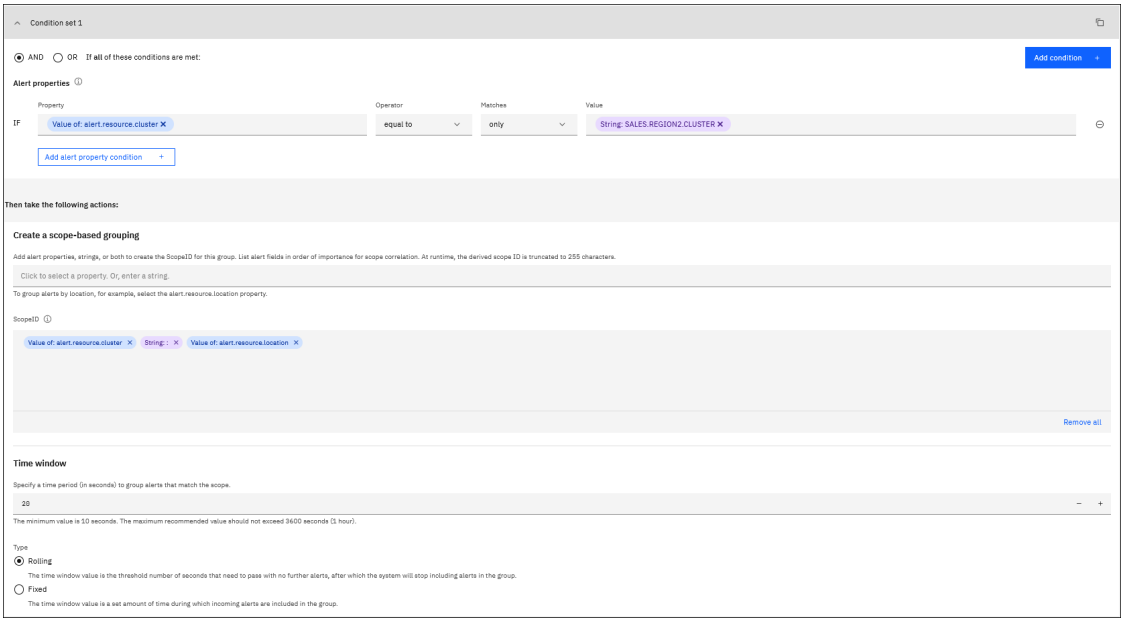


Figure. Scope-based policy

9. Specify a **Time window** (in seconds) to group alerts that match the scope. The minimum value is 10 seconds. The maximum recommended value should not exceed 3600 seconds (1 hour).

   Select the time window **Type**:

   - **Rolling**: The time window value is the threshold number of seconds that need to pass with no further alerts, after which the system stops including alerts in the group.
   - **Fixed**: The time window value is a set amount of time during which incoming alerts are included in the group.

10. Click **Create policy**.

Scope-based alert grouping is activated for alerts that match the condition, based on the ScopeID specified.

New and updated policies can take up to 1 - 3 minutes to take effect.

Assign a runbook to alerts

Assign a runbook to alerts for easier automated resolution.

Important: in runbook policies, the actions from every matching condition set are executed. Policies are triggered upon alert creation only.

About this task

In this example, you want to have a runbook take action when you receive alerts warning that your disk space is filling up. The runbook could delete the contents of the `/tmp` directory to free up space. You can create a policy to use this runbook every time alerts warn of your disk space becoming full.

Note: This example assumes you have a runbook that is called **Clear tmp directories**. For information about setting up runbooks, see Runbooks.

Example

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Create policy**.
4. Click the **Assign a runbook to alerts** tile.
5. Enter a name in **Policy name**, for example, "Clear tmp directories". You can also add an explanation of the policy in **Description** to help you and others understand the purpose of the policy, for example, "Policy to assign a runbook to alerts that indicate that disk is full due to /tmp filling up".
6. Set the **Execution order** to 50.
7. Define how the policy is triggered to assign a runbook to alerts. Once triggered, it will look for alerts that match the conditions you specify in order to take action. The policy triggers are when an alert is created, updated, or both. Updated means the alert state changes. For example, if an alert changes from Severity 1 to Severity 5. In this example, select **An alert is created**.
8. Define the following conditions for alerts that will assign a runbook to alerts:
    1. Click **Add condition** and select **Alert property**.
    2. From the Property drop-down list, select **alert.type.classification**. You can type "class" and the system will show in the property drop-down list all alert properties that contain the text "class", which in this case is only alert.type.classification. From the Operator drop-down list, select **equal to**. From the Matches drop-down list, select **only**. In the Values field, enter **Disk Full** and select **String:Disk Full**.
    3. Click **Add alert property condition**.
    4. From the Property drop-down list, select **alert.severity**. From the Operator drop-down list, select **greater or equal**. From the Matches drop-down list, select **only**. In the Values field, select **5-Major**.
    5. Click **Add alert property condition**.
    6. For the third condition, select **summary** from the Property drop-down list. From the Operator drop-down list, select **contains**. From the Matches drop-down list, select **only**. In the Values field, enter **/tmp** and select **String:/tmp**.
9. In **Assign Runbook(s)**, select one or more runbooks from the list to assign. In this example, select the **Clear tmp directories** runbook.
10. Review the list of parameters that are required for each runbook selected. You must decide how the values of the runbook parameters are filled by creating a parameter mapping. Select from the following mapping options:
    - **Choose from alert**: Select the name of an alert field from the tree view, its value will be passed to the runbook parameter. **Note:** The parameter **Type** is displayed on the UI. If this option is selected, you should select alert fields that contain matching types of data. For example, strings vs integers.
    - **Enter static value**: A fixed value for the parameter at runtime.
    - **Choose at runtime**: After launching the runbook from the incident page, if this option is selected, you must provide a value at runbook execution time. Sometimes the default value as set in the runbook is prefilled for the parameter, in this case just edit the value at execution time.
    - **Use default parameter value**: If the runbook defines a default value for this parameter, you can use that value. This option is greyed out if no default was set in the runbook.

For this example, select "Use default parameter value".

Example runbook policy

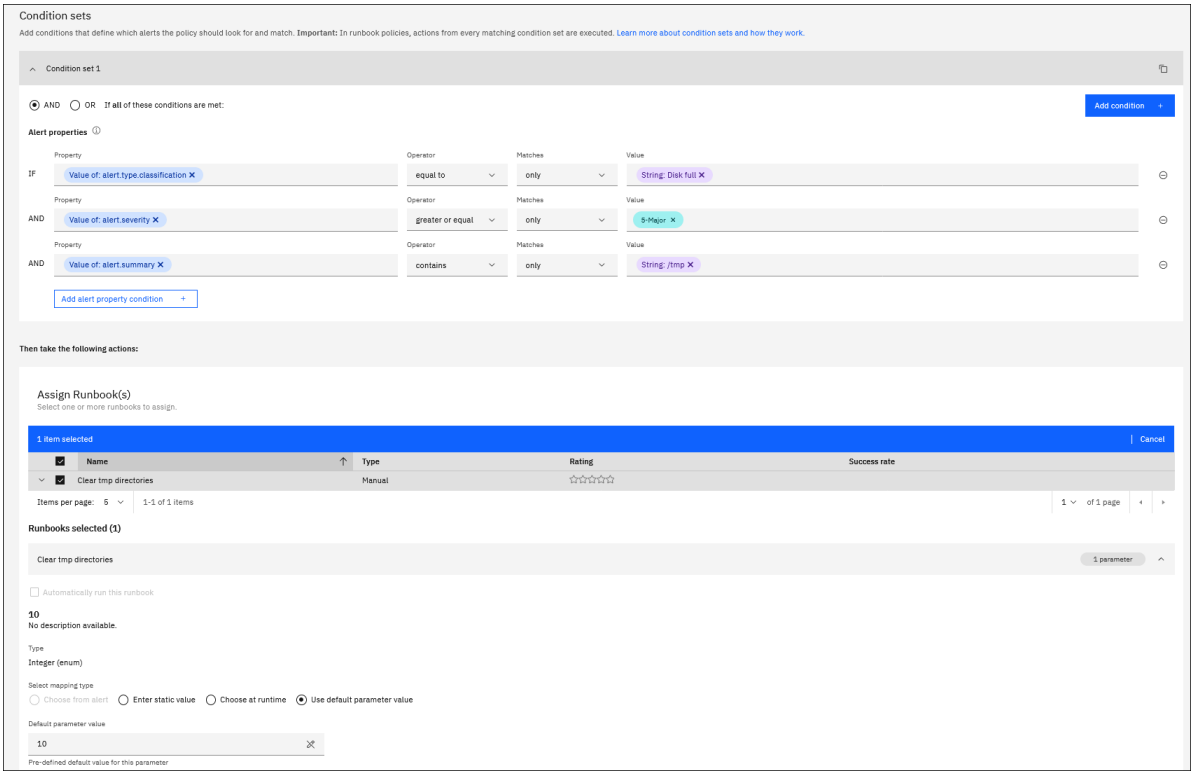


Figure. Supression policy

11. Click **Create policy**.

New and updated policies can take up to 1 - 3 minutes to take effect.

Note: If more than one runbook policy is triggered for the same alert and the same runbook is assigned, the policy parameter selections from the last matching condition set or policy are passed to the runbook.

Promote alerts to an incident

Create actionable incidents from alerts to improve insights into your issue. You can then automatically assign incidents to a user group and owner. You can also send notifications about the incident through ChatOps.

Important: In incident policies, only the actions from the first matching condition set are executed. Policies are triggered upon alert creation only.

About this task

In this example, all high priority incidents that include alerts at or above a defined business criticality level are automatically assigned to the Db2 admin group with a notification sent to the group's team leader of such incidents.

There are three preset business criticalities available in Cloud Pak for AIOps:

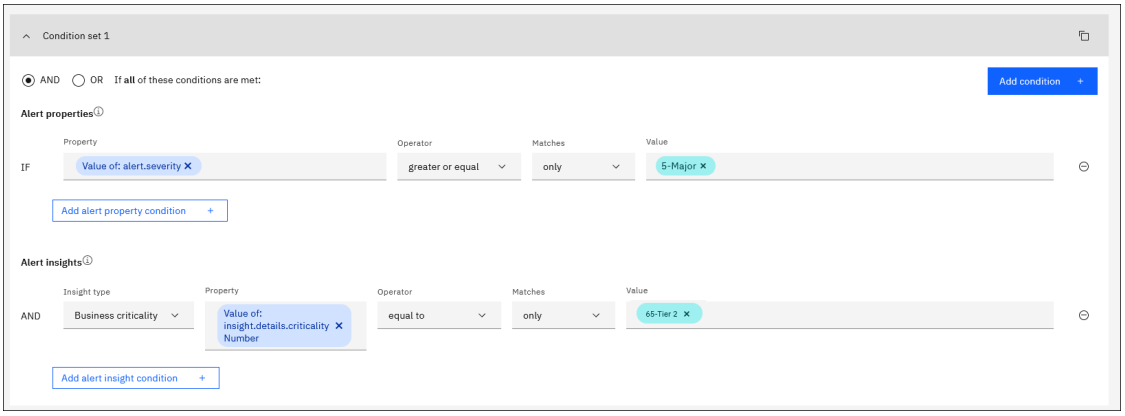
Table. Preset business criticalities

Name	Description	Criticality score
Tier 1	Highest priority	100
Tier 2	Medium priority	65
Tier 3	Lowest priority	25

Note: Preset business criticalities can be edited after they are added to your environment. For more information, see Defining business criticality.

Example

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Create policy**.
4. Click the **Promote alerts to an incident** tile.
5. Enter a name in **Policy name**, for example, "Assign high priority Db2 incidents to Db2 admins". You can also add an explanation of the policy in **Description** to help you and others understand the purpose of the policy, for example, "Automatically assign any priority 1 incidents from Db2 hosts to the Db2 admin group, and notify team leader".
6. Set the **Execution order** to 1.
7. Define how the policy is triggered to create an incident. Once triggered, it will look for alerts that match the conditions you specify in order to take action. The policy triggers are when an alert is created, updated, or both. Updated means the alert state changes. For example, if an alert changes from Severity 1 to Severity 5. In this example, select **An alert is created**.
8. Define the following conditions for alerts that will trigger the creation of an incident:
    1. Click **Add condition** and select **Alert property**.
    2. From the Property drop-down list, select **alert.severity**. You can type "sev" and the system will show in the property drop-down list all alert properties that contain the text "sev", which in this case is only alert.severity. From the Operator drop-down list, select **greater or equal**. From the Matches drop-down list, select **only**. In the Values field, select **5-Major**.
    3. Click **Add condition** and select **Business criticality**.
    4. The Property field contains the business criticality value by default. From the Operator drop-down list, select **greater or equal**. From the Matches drop-down list, select **only**. In the Values field, select **65-Tier 2**.



Incident policy condition set

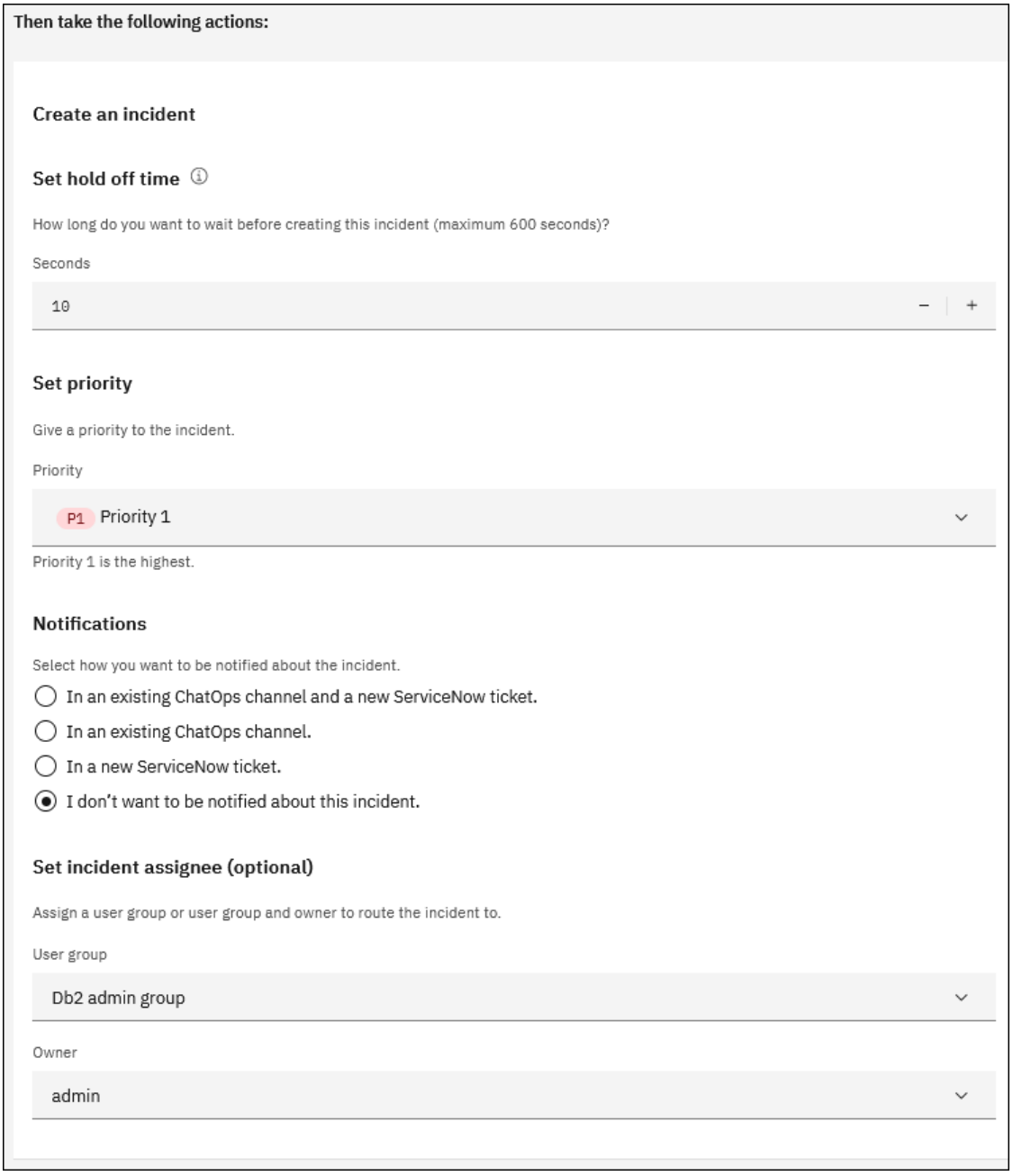
9. **Set hold off time**. Delaying the creation of an incident provides time for data about the issue to be ingested. This can allow the system to make a better decision about what incidents should be created. For example, for flapping alerts or alerts which might be resolved by an automated runbook. In these cases, the incident would only be opened if the alert remains open beyond the hold off time. Eliminating incidents from the UI which ultimately are not actionable.

    Enter the number of seconds that you want to wait before the incident is created (maximum 600 seconds). In this example, the hold off time is 10 seconds.
10. **Set priority**. Give a priority level to the incident that is created when the policy conditions are met. For this example, select **Priority 1**.
11. Select how you want to receive **Notifications** about the incident:

- In an existing ChatOps channel and a ServiceNow ticket.
- In an existing ChatOps channel.
- In a new ServiceNow ticket.
- I don’t want to be notified about this incident.

12. Optionally, assign a user group and owner to route the incident to. In this example, the user group is the Db2 admin group, and the owner the group's team leader.

    Example incident policy actions:



Incident policy actions

13. Click **Create policy**.

New and updated policies can take up to 1 - 3 minutes to take effect.

Invoke an IBM Tivoli Netcool/Impact policy

Enable Cloud Pak for AIOps to invoke an IBM Tivoli Netcool/Impact policy that will take actions on your alerts.

Note: If you have existing **Impact Policy Language (IPL) policies** (versus JavaScript policies) that you want to use in Cloud Pak for AIOps to return the event, you must modify the return statement from:

```
ReturnEvent(EventContainer);
```

To:

```
AIOPS_IPL_Utils.AIOPSReturnEvent(EventContainer);
```

For example:

```
@Summary=EventContainer.mynewsummary + " Enriched with this message";

AIOPS_IPL_Utils.AIOPSReturnEvent(EventContainer);
```

About this task

For example, you want Cloud Pak for AIOps to invoke an existing IBM Tivoli Netcool/Impact policy called Notify_Customer_Email. The policy accepts incoming event information from an event reader service and uses this information to look up affected internal customers in an external database. The policy then sends an email to each customer and reports the event data to a third-party help desk system that uses the web services API provided by that application.

For more information about setting up IBM Tivoli Netcool/Impact policies, see the IBM Tivoli Netcool/Impact documentation.

Important:

- To successfully enable your IBM Tivoli Netcool/Impact policy with Cloud Pak for AIOps, you must first configure an IBM Tivoli Netcool/Impact integration in **Define** > **{{site.data.keyword.ui_data_integrations}}**.
- By default, IBM Tivoli Netcool/Impact policies are invoked without input or transformation. Editing options are available in the policy template.

Example

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Create policy**.
4. Click the **Invoke Netcool/Impact** tile.
5. Enter a name in the **Policy name**, for example, "Notify customer help desk". You can also add an explanation of the policy in the **Description** to help you and others understand the purpose of the policy, for example, "Look up customers affected by the event in the external database."
6. Set the **Execution order** to 50.
7. Define how the policy is triggered to invoke an IBM Tivoli Netcool/Impact policy. Once triggered, it looks for alerts or incidents that match the conditions you specify in order to take action. First, select the entity that this policy acts on. Choose either **Alert** or **Incident** properties. The default trigger entity is Alert.

   The policy triggers are when an alert or incident is created, updated, or both. Updated means the alert or incident state changes. For example, if an alert changes from Severity 1 to Severity 6, or if an incident escalates from Priority 5 to Priority 1 (this is an escalation as Priority 1 is the highest priority level).

   Note: Switching between alert and incident entities resets the input fields for triggers, condition sets, and actions.

   For this example, we select **After an alert is created**.
8. Define the following conditions for alerts that trigger your IBM Tivoli Netcool/Impact policy:

   1. Click **Add condition** and select **Alert property**.
   2. From the Property drop-down list, select **alert.severity**. You can type "sev" and the system will show in the property drop-down list all alert properties that contain the text "sev", which in this case is only alert.severity.

From the Operator drop-down list, select **greater or equal**. From the Matches drop-down list, select **only**. In the Values field, select **5-Major**.

9. In **Invoke Netcool/Impact**, specify the IBM Tivoli Netcool/Impact integration instance to invoke the policy. If you don't see anything in the **Impact connection** dropdown list, see IBM Tivoli Netcool/Impact integrations.

10. Select your **Impact policy** Notify_Customer_Email from the drop-down list. You can start typing the policy name in the field that is provided and the matching IBM Tivoli Netcool/Impact policy names are suggested.

11. By default, your IBM Tivoli Netcool/Impact policy parameters are mapped to all alerts or all incidents. Select from the following IBM Tivoli Netcool/Impact policy parameter-mapping options:

    - Send the full alert or incident
    - No input
    - Customize

Send the full alert or incident

Send the entire alert or incident without alteration. Click **Create policy** to save the policy.

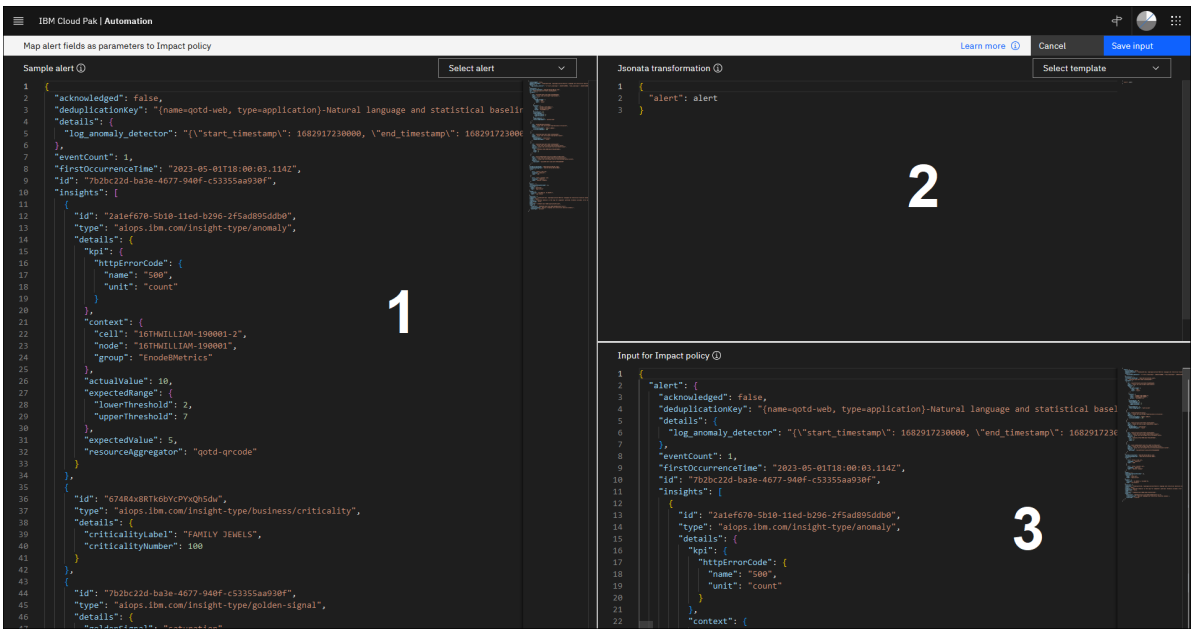
No input

Your IBM Tivoli Netcool/Impact policy requires no input. Click **Create policy** to save the policy.

Customize

Select this option if you want to customize alert or incident field parameters that are used as input for your IBM Tivoli Netcool/Impact policy. Click **Edit parameter mapping** to continue.

In this example, alert fields are being mapped as parameters to an IBM Tivoli Netcool/Impact policy. If incidents are selected as the policy trigger entity (see step 7 in the preceding example), then incident fields are mapped as parameters here. The editor is divided into three sections. You can click and drag the vertical and horizontal middle lines to resize the sections.



Edit parameter mapping

- Section 1: **Sample alert**

  Use the **Select alert** drop-down menu to load a sample alert, or go to the Alert Viewer to copy and paste a raw alert by opening **Alert details** > **Information** > **Raw** > **Copy to clipboard**. For more information, see Alert details.

  The sample alert provides variables available for use in your mapping, including the alert that the policy is acting on. You can modify the variables to test out different possible inputs to your IBM Tivoli Netcool/Impact policy, for example, by overriding the alert with one from your alert list.

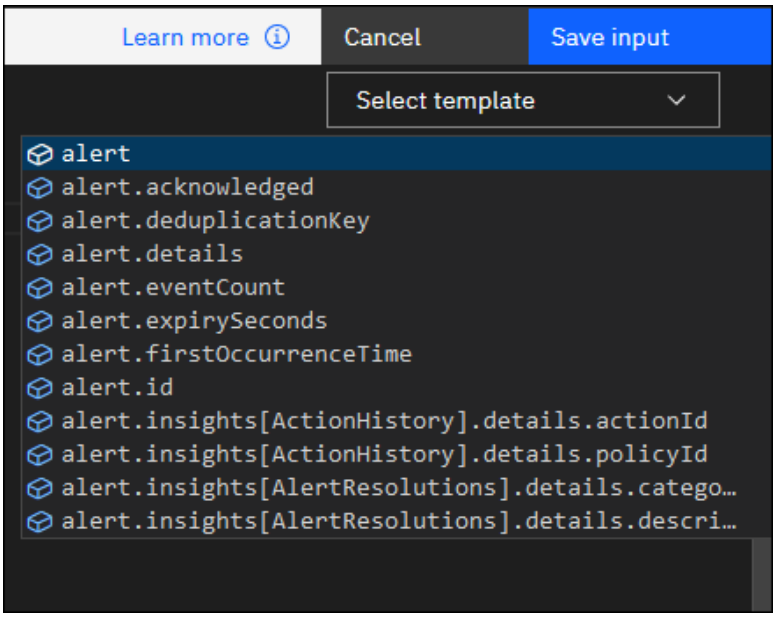
  Note: The sample alert is only used to test your transformation and has no bearing on the execution of your policy.

- Section 2: **JSONata transformation**

  A JSONata expression translates between the variables available to the Cloud Pak for AIOps policy and the EventContainer object that is made available to the IBM Tivoli Netcool/Impact policy during execution. Use the JSONata transformation section to include your own arguments for an IBM Tivoli Netcool/Impact action by making a selection from the **Select template** drop-down menu. If you are invoking an IBM Tivoli Netcool/Impact policy that was originally written for an IBM Tivoli Netcool/Impact EventReader, you will likely want to translate the alert variable to a set of ObjectServer columns here.

  When you start typing in **JSONata transformation** (or press Ctrl + Space), a suggestion panel with alert properties is shown. Move the cursor into the panel to scroll and select from the list of properties. As you make changes in this section, the **Input for Impact policy** (section 3) shows a preview of your transformation.

  Note: Policies support individual details by name. However, there is no **`alert.details.detailsname`** displayed in the JSONata transformation list of properties. To pass such a value, select **`alert.details`** and then add **`."details_name"`**. The **`alert.details`** name must be specified in quotation marks (**" "**). That is **`alert.details."details_name"`**. The value of **`alert.details."details_name"`** will then be passed at run time.

  Parameter suggestion panel

  Warning: Any changes that you make to the source alert are reset if you reedit the parameter mapping.

- Section 3: **Input for Impact policy**

  This section shows the results of your transformation and what the content of the EventContainer variable in IBM Tivoli Netcool/Impact will contain, based on the current sample alert and transformation rules that are used in section 2.

12. Click **Save input** to save your changes and close the editor. Then, click **Create policy** to save the policy.

    New and updated policies can take up to 1 - 3 minutes to take effect.

Supported Anomaly insights properties

In a policy condition set, you can suppress alerts and create reports from alerts by using the following types of **Anomaly insights** properties:

Table. Anomaly insights properties

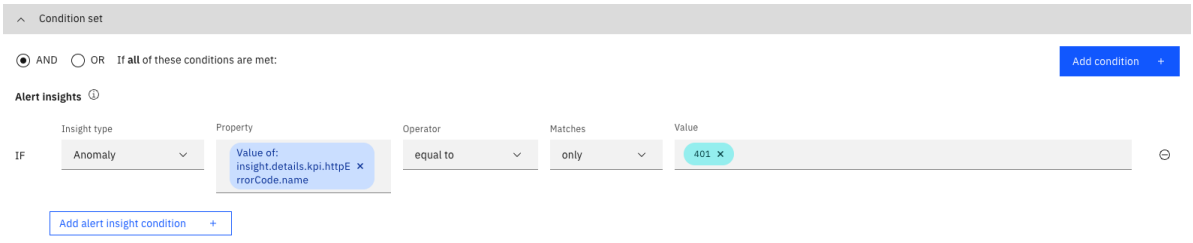
Property	Description	Type	Possible values

Property	Description	Type	Possible values
`httpErrorCode.name`	HTTP error code	`string`	400, 401, 403, 408, 429, 500, 502, 503, 504, 5xx
`httpErrorCode.unit`	Unit for the HTTP error code KPI	`string`	Count
`exception.name`	Name of the exception (for example, a token that contains the string `exception` found within a Log)	`string`	
`exception.unit`	Unit for the exception KPI	`string`	Count
`messageId.name`	An IBM WebSphere MQ message ID	`string`	
`messageId.unit`	Unit for the message ID KPI	`string`	Count
`metric.name`	Metric name	`string`	
`metric.unit`	Unit for the metric KPI (for example, bytes, mb, ms, s)	`string`	
`actualValue`	The observed value for a KPI.	`number`	
`expectedlValue`	The expected value for a KPI.	`number`	
`expectedRange.upperThreshold`	The expected upper bound of the count/frequency of the KPI in a given time aggregated window.	`number`	
`expectedRange.lowerThreshold`	The expected lower bound of the count/frequency of the KPI in a given time aggregated window.	`number`	
`resourceAggregator`	This field is optional. An entity in the underlying client application that can be used to logically aggregate all anomalies that are related to the same application resource. For example, in the context of Metric anomalies this would be a server that might be used to aggregate anomalies across all ports exposed by that server. For Log anomalies, this might be a Kubernetes ReplicaSet that aggregates all the pods it manages.	`string`	

Using properties for alert suppression

Scenario 1: Alert suppression based on HTTP error codes

You might expect the system to generate certain HTTP error types at regular intervals. However, these alerts do not require action and you do not want alerts and incidents generated based on these errors. You can enable alert suppression to avoid new alerts from being generated.



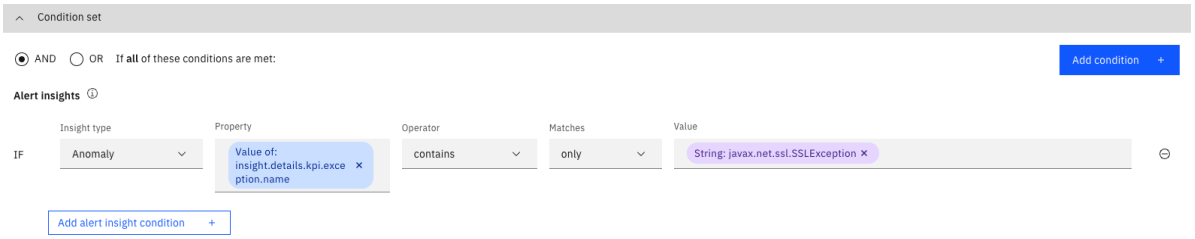
Example 1: Suppress alerts for any 401 error code.

Scenario 2: Alert suppression based on exceptions present in log messages

Some exceptions present in log messages might not be a priority or cause any outages for your environment. As a result, you might want to suppress any alerts that are generated by these exceptions captured in logs.

Log anomaly suppression policies can reference only computed or derived entity values from the RSA algorithm, such as the exception name. To suppress certain log anomaly alerts based on the exception name, go to the IBM Cloud Pak® for AIOps Suppress Alerts automation page. Then, from the Alert insight category suppression options, choose the

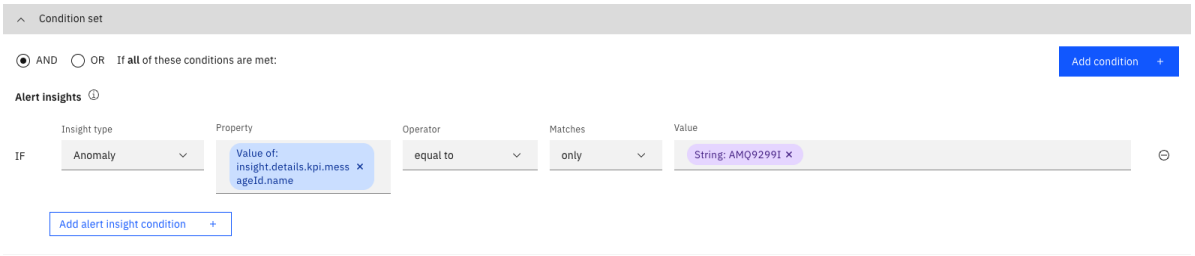
insight.details.kpi.exception.name property. However, you cannot define a log anomaly suppression policy that references the string message that is found in the log record.



Example 2: Suppress all alerts generated by exceptions of the type javax.net.ssl.SSLException.

Scenario 3: Alert suppression based on domain-specific error code entities that are present in the logs

Domain-specific logs, such as IBM MQ or WebSphere, include error code entities that you can use to set policies to suppress alerts. For example, use the **insights.details.kpi.messageId.name** field in an Anomaly insights condition set to suppress alerts that match that entity.

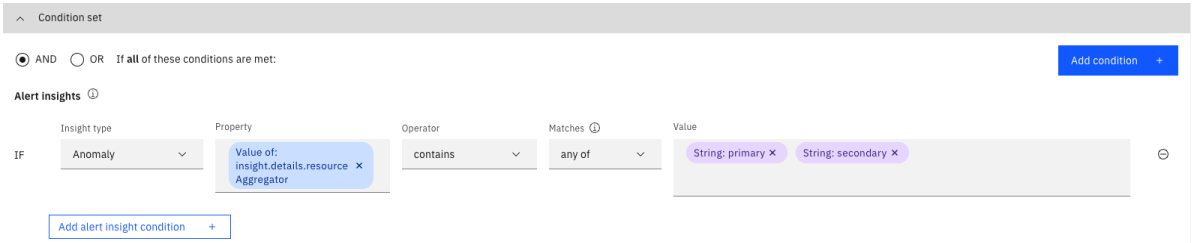


Example 3: Alert suppression based on an incident where the message name equals AMQ9299I.

Using properties for creating runbooks

Scenario 4: Create a runbook based on resource name

Use the **resourceAggregator** field in an Anomaly insights condition set to create a runbook based on resource name.

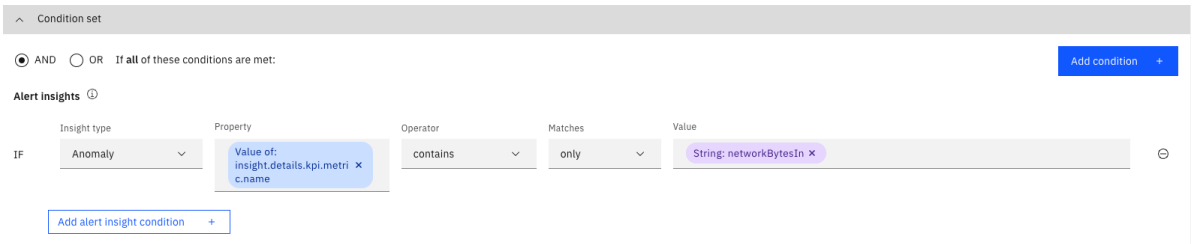


Example 4: Create a runbook for any resource that contains either primary or secondary.

Using properties to invoke IBM® Tivoli® Netcool®/Impact

Scenario 5: Invoke Netcool/Impact based on metric name

Use the **metric.name** field in an Anomaly insights condition set to invoke Netcool/Impact based on metric name.



Example 5: Invoke Netcool/Impact where the metric name contains networkBytesIn.

Policy editor

You can create or customize policies by adding or editing conditions in the policy and the actions that are triggered. Policies are triggered upon alert creation only. This topic contains an example of editing a policy from the policy table. You can also edit a policy from the **Details** tab in the side panel.

Important: In incident policies, only the actions from the first matching condition set are executed. In runbook policies and scope-based policies, the actions from every matching condition set are executed. Suppression policies contain only one condition set.

Policy details and Execution order

The first two sections of the policy editor are **Policy details** and **Execution order**. Here you name (or rename) the policy and add a description. To set the execution order for the policy, use the slider or enter a value between 1-100. Policies with a lower number are processed first. Preset policies that handle system needs are executed before or after those with a value of 1-100. User-created suppression policies are always executed first. The order is set to 0 and cannot be changed.

To see how your current policies are set, click on the number of policies listed below each range.

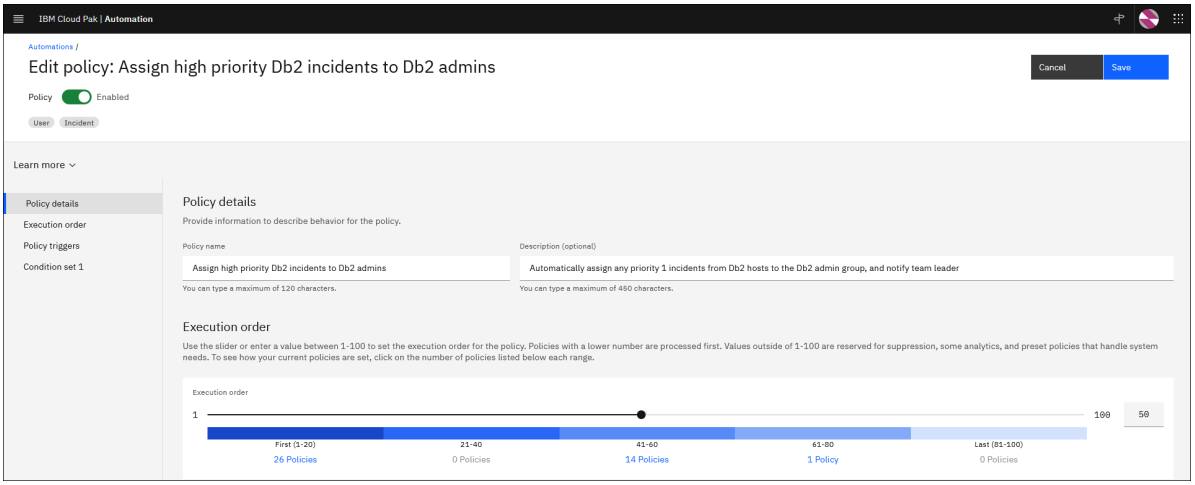


Figure. Policy details

Policy triggers

With the exception of suppression policies, all other policy types have a **triggers** section. These policies are triggered when an alert is created, updated, or both. Updated means the alert state changes. For example, if an alert changes from Severity 1 to Severity 5. If the policy is triggered by the alert create or update, then the conditions are consulted.



Figure. Policy triggers

Condition sets

In a **Condition set** you configure the conditions that alerts must meet. For new policies, start by adding a condition set. At least one is required. Condition sets are created based on alert properties and Cloud Pak for AIOps alert insights. Click **Add condition** to select a new condition type from the drop-down menu.

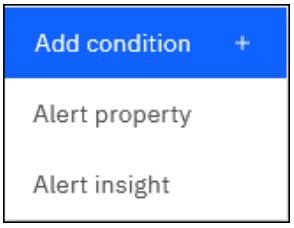


Figure. Custom alert insights

Choose from the following condition types:

- **Alert properties**: alert attributes that are predefined for IBM Cloud Pak for AIOps and common to most alerts. For more information, see Alert details.
- **Alert insights**:
  - *Anomaly*: Log anomaly and metric anomaly attributes that are predefined for IBM Cloud Pak for AIOps. For more information, see Supported Anomaly insights.
  - *Business criticality*: Set in Resource management to define the importance of an application, resource group, or resource to the business. For more information, see Defining business criticality.
  - BETA *Feedback:* Defines whether a user agrees that an alert is valid or not as expressed through user input in the Alert details. After enough feedback is collected, a Feedback Learning model is trained and deployed. Selecting whether an alert is valid or invalid is available only for technology preview, and is not for production usage. For more information, see Alert details.
  - *Golden signal*: Select from predefined golden signal `labels` and `types`.
    - *Labels*: `traffic`, `latency`, `error`, `saturation`, `availability`, `information`.
    - *Types*: `effect`, `cause`, `none`.
  - *Seasonal*: Alerts that occur within a seasonal time window.

Note: Schema descriptions for both Alert properties and Alert insights are displayed on the UI. Click anywhere in **Property** field and move the cursor over the properties to display the descriptions.

Schema descriptions

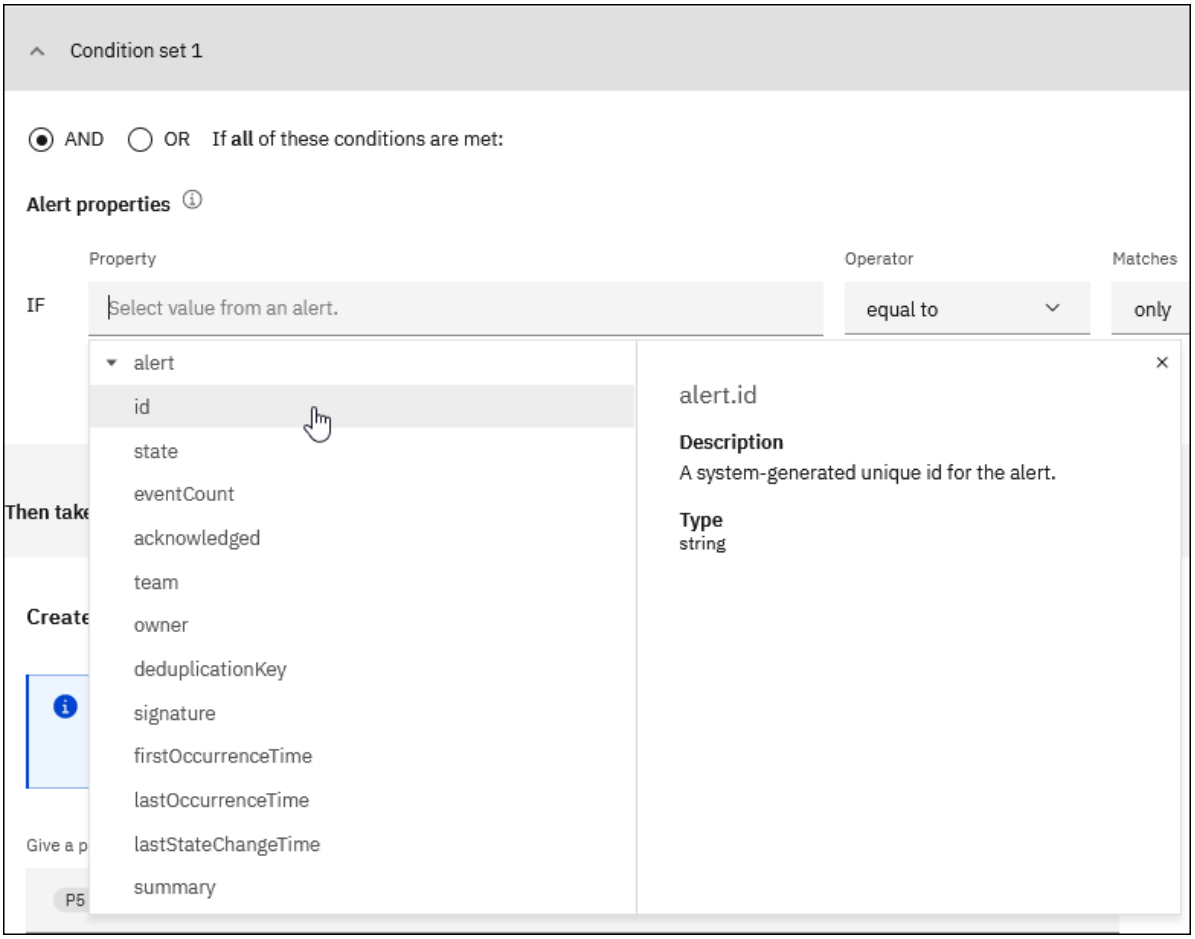


Figure. Schema descriptions

When adding conditions, you can join multiple condition types by using the AND and OR operators. The AND operator means that alerts are matched only if all of the individual conditions are true. The OR operator means that alerts are matched if any of the individual conditions are true.

In the fields provided, select the **Property**, **Operator**, **Matches**, and **Value** for the new condition. Setting a **Value** field to the value of another property allows you to compare the value of one property to one or more other properties. Click anywhere in the **Value** field and scroll the tree-view to select from the available options. You can also enter the last part of the fully qualified tree-view name to see and select the tree-view "value of" alert or insight field. To enter a string value, type the string (e.g. "Error") in the **Value** field and then select **String: Error** from the drop-down. If you enter a fully qualified tree-view value, you can save it as a string versus the "value of".

If you need more than one condition set, click **Add condition set**. Alternatively, to copy and paste an existing condition set click **Copy condition set**. To rename a condition set, point the cursor over the condition set name and click Edit. You can use the sidebar on the left of the window to navigate a policy's condition sets and actions. To remove a condition set, click **Delete**. At least one condition set is required.

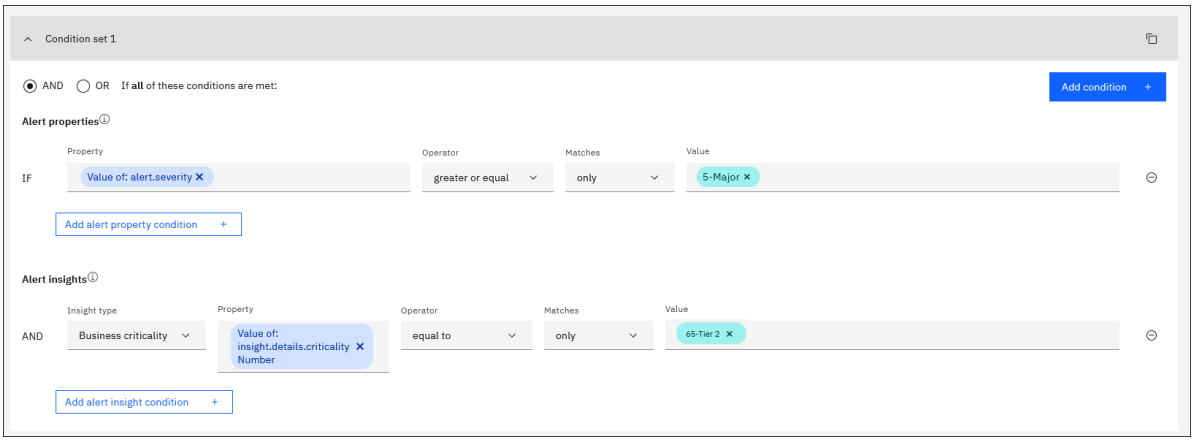


Figure. Condition set

Note:

- Suppression policies contain only one condition set. You cannot add or copy condition sets in these policies.
- Unlike **Alert details** properties, **Alert insights** conditions like **Business criticality** and **Anomaly** conditions cannot evaluate **`true`** where Business criticality or Anomaly insights contain no value. For these two property types, Cloud Pak for AIOps checks first for a value, and if it is not present, **`false`** is returned. If the value is present, the condition that is assigned by the user is evaluated, and **`true`** or **`false`** is returned as determined. Therefore, it is a best practice to avoid use of the negative **Operator** property "**not equal to**" when you use Business criticality and Anomaly insights properties.

Actions

Actions are where you specify what is triggered when the policy conditions are met:

- Suppression policies: the action is always to suppress the alerts described in the condition sets.
- Scope-based policies: the action that is taken is defined by alert properties and strings to create the ScopeID for a group. These properties define the scope for how the alerts are grouped.
- Runbook policies: choose one or more runbooks to assign when the policy conditions are met, and select how parameters are passed.
- Incident policies: the action is the creation of an incident with a specified priority level (example below). You can optionally delay the creation of an incident by setting a hold off time, select how you want to be notified about the incident, and assign a user group or user group and owner.
- Invoke an IBM Tivoli Netcool/Impact policy: enable Cloud Pak for AIOps to invoke an IBM Tivoli Netcool/Impact policy that will take actions on your alerts.

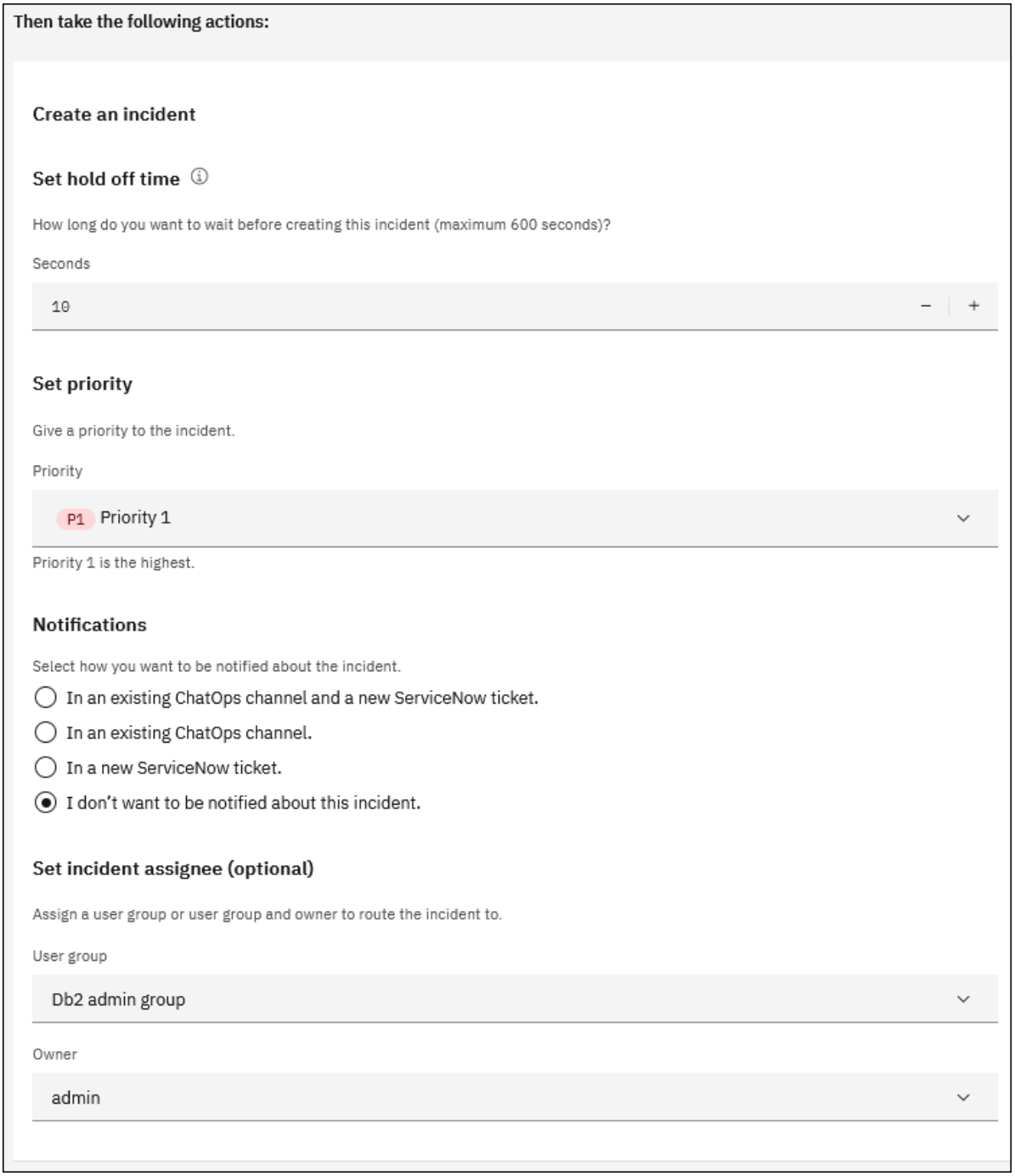


Figure. Actions

Example

In the following example, a condition is added to an existing policy so that the policy applies only to alerts that have a prefix of either "Error" or "Warning" in their Summary field.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. In the table row of the policy that you want to edit, click the menu overflow icon (the three dots at the end of the row) to and select **Edit policy**.
4. Click **Add condition** and select **Alert property**.

1. From the Property drop-down list, select **alert.summary**. You can type "sum" and the system will show in the property drop-down list all alert properties that contain the text "sum", which in this case is only alert.summary.
2. From the Operator drop-down list, select **Contains**.
3. From the Matches drop-down list, select **any of**.
4. In the Values field, type **Error** and then click **String:Error** in the pop-up. In the same field, type **Warning** and click **String:Warning**.



Figure. Policy example

5. Click **Save** to save your changes.

   If the save button is not active, one or more items might be missing in the conditions or runbook parameter sections. If clicking save fails to exit the edit session, again, check that no required details are missing.

Note: New and updated policies can take up to 1 - 3 minutes to take effect.

IBM Cloud Pak® for AIOps determines which of your alerts were caused by the same problem. If an active incident exists for a given problem, a new incident is not created.

For more information about creating policies, see the following topics:

- Suppress alerts
- Group alerts based on scope
- Assign a runbook to alerts
- Promote alerts to an incident

Best practices for condition sets

For certain scenarios, some best practices are recommended for the **Operator** and **Matches** fields in a policy condition set.

Scenario: comparing against multiple string values

- For the operator **not equal to** use **all of**. If the property under test is null a false positive will occur, therefore, also add a check for **not empty** property.
- For the operator **does not contain** use **any of**. If the property under test is null a false positive will occur, therefore, also add a check for **not empty** property.
- For the operator **contains** use **any of**.

Scenario: looking for unassigned owners

- If you need a policy condition to search for unassigned owners or teams, the **is empty** operator does not work with the `alert.owner` or `alert.team` properties. Use the **equal to** operator instead, with **only** and enter a minus sign (-) value.



Example: search for unassigned owners

Refreshing the policies table

Refresh the policies table to view all the latest policies at the current point in time.

If any filters are applied to the table, click **Clear filters** first to view a complete list of policies.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click Refresh to refresh the table at anytime. A refreshed policy count is displayed above the table beside **Policies**.
4. You can also click start to refresh the table with with auto-refresh every 90 seconds. Click stop to turn off auto-refresh.

Note: Applying filters automatically refreshes the table.

Managing multiple policies

You can multiselect policies and apply actions to all of your selections at one time.

Note: When a multiselect of policies includes different policy types, not all actions can be applied. For example, preset policies cannot be deleted.

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Select multiple policies from the table by using the checkboxes at the beginning of the table rows.

   Tip: You can select all of the policies in the table by clicking the checkbox in the table header beside **Policy name**.
4. A banner above the table displays the number of policies that are selected, together with a menu of options. The following actions can be applied to multiple policies at the same time:

   - **Enable**
   - **Disable**
   - **Delete**

Renaming policies

A global unique identifier string is used to identify analytics policies. You can rename policies to give them more meaningful names. Preset policy names cannot be customized.

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Point the cursor over the **Policy name** that you want to change and click Edit.
4. Enter a new policy name in the field provided.
5. Click the check mark to save the policy name or click cancel to discard your changes.

Note: When editing is enabled for the policy, the name can also be updated in the policy editor.

Archiving policies

You can archive policies that are not in use. Archived policies don't act on incoming alerts.

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Filter the table to display **Temporal** and/or **Seasonal** policies, or search for a policy by entering a search term in the field provided.
4. In the table row of the policy that you want to archive, click the options menu (three vertical dots) at the end of a table row and select **Archive policy**.
5. To confirm the archive, click **Archive policy**.

Archived policies can be viewed and managed in Draft and archived policies.

About runbooks

Use Runbook Automation to build and execute runbooks that can help IT staff to solve common operational problems. Runbook Automation can automate procedures that do not require human interaction, thereby increasing the efficiency of IT operations processes. Operators can spend more time innovating and are freed from performing time-consuming manual tasks.

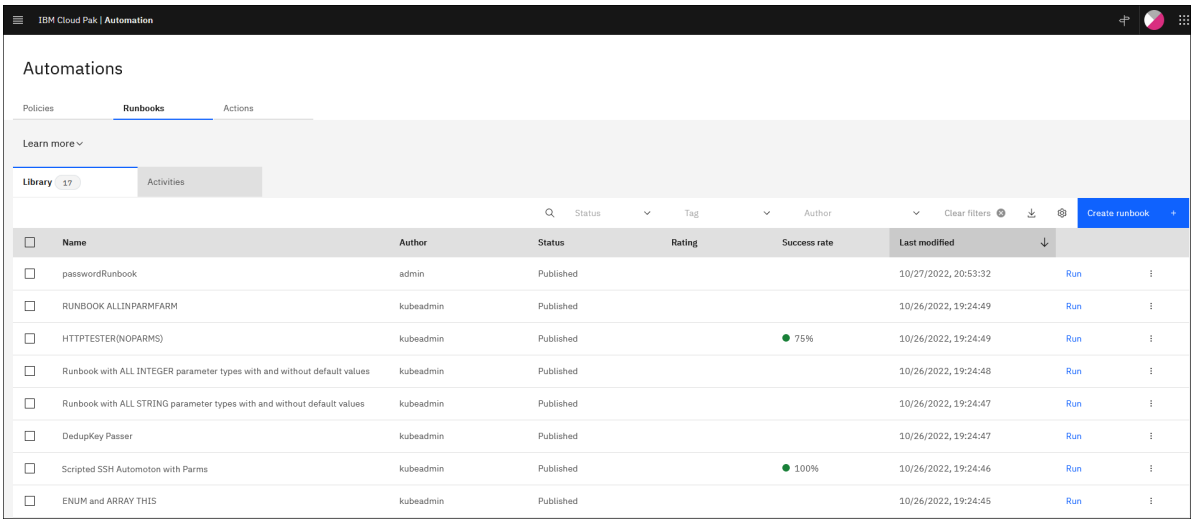


Figure. Runbooks Library UI

- What problems does Runbook Automation address?

  IT systems are growing. The number of events are increasing, and the pressure to move from finding problems to fixing them is increasing. Runbook Automation supports operational and expert teams in developing consistent and reliable procedures for daily operational tasks.

- How can I simplify daily operational tasks?

  Using Runbook Automation, you can record standard manual activities, so that they are run consistently across the organization. The next step is to replace manual steps with script-based tasks.

- What is a runbook?

A runbook is a controlled set of automated and manual steps that support system and network operational processes. A runbook orchestrates all types of infrastructure elements, such as applications, network components, or servers.

Runbook Automation helps users to define, build, orchestrate, and manage runbooks. Metrics about runbooks are also provided. A knowledge base is built up over time through collaboration between Subject Matter Experts (SMEs) and operational staff.

Runbook types

Transfer your procedure based documentation to manual runbooks. Then add actions and continue with semi-automated runbooks or fully automated runbooks.

Runbook Automation supports three different types of runbooks:

- Manual runbooks

  Steps describe the exact procedure that an operator must follow. The operator uses standard tools, which can be accessed from their working environment.

- Semi-automated runbooks

  Each step describes exactly what an operator must do. Additionally, the operator can run an automated task on a target system.

- Fully automated runbooks

  The runbook runs automatically without operator interaction.

Actions

Actions are a task that run autonomously on a target system. For example, a shell script that identifies large files in a file system.

Runbook Automation supports the following technologies to connect to a backend system:

- Script

  Actions of this type contain a complete script. When a SSH Action runs, the content is sent to the target system and executed. Scripts can be of type Bash or PowerShell, depending on the target system.

- HTTP

  Actions of this type define the specific properties to make a secure HTTP call.

- Ansible

  Actions of this type reference existing Job Templates or Workflow Job Templates on an Ansible Automation Platform Controller server and run them with the defined variables.

For more information, see Actions.

Runbooks Library

View the available runbooks, find matching runbooks for the alerts that you need to resolve, and run the runbooks. You can also review the runbooks that you have used to date. The library also provides ready-to-go sample runbooks. Load the examples to explore what you can do with runbooks. Document your existing operational procedures using the runbook editor. Share the knowledge and experience that you documented in a runbook with your team members. The runbook editor provides

editing capabilities as well as runbook elements that help to document operational procedures. Runbook elements are designed to avoid mistakes.

For more information about the Runbooks Library, see the following topics:

- Run your first runbook
- Load example runbooks
- Create a runbook
- Edit a runbook
- Configuring runbook creation
- Runbook versions
- Save and publish a runbook
- Filtering runbooks
- Assigning runbooks to groups
- Export and import runbooks
- Delete a runbook

Run your first runbook

If you are using Runbook Automation for the first time, it is recommended to load example runbooks and then edit and publish a runbook. For more information, see Load and reset example runbooks and Edit a runbook.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Runbooks**.

   The **Library** tab is displayed by default.
4. Select a runbook and click the menu overflow icon (the three dots at the end of the row) and select **Preview** to find out what a runbook consists of:

   - **Procedure**

     The Procedure section displays the runbook steps. For more information about the runbook elements, see Edit a runbook.

   - **Information**

     The information tab provides descriptive details of the runbook and contains the following fields:

     - Description: shows a description of the runbook.
     - Runbook ID: ID of the runbook which can be used to identify the runbook using the Runbook Automation HTTP REST Interface.
     - Created by: the user who created the runbook.
     - Last modified at: the date and time the runbook was last modified.
     - Last modified by: the user who last modified the runbook.
     - Type: indicates the runbook type, that is manual, semi-automated, or fully automated.
     - Tags: categorization tags associated with the runbook.
   - **Versions**

     Shows all draft, published, and archived versions of the runbook. You can also see the user that created each version and the creation date.
5. Back on the main **Runbooks** page, select one of the runbooks and click **Run**.
6. The **Run runbook** page opens, which consists of a procedure description on the left side of the window and a **Details** and **Information** section on the right. **Details** contains the following collapsible sections:

   - **Parameters** – displays a list of the parameters used.
   - **Actions** – displays a list of the actions used.

7. In the **parameters** section you are requested to enter parameter values. Click the information icon to see a description and a reference to where this parameter is used within the runbook. Enter the parameter values, for example host1 for the parameter **HOSTNAME** and click **Start runbook**.

   Note:

   - In some cases, you might not know the parameter value when starting the Runbook. For example, the parameter could be determined by a runbook step during the runbook execution. In this scenario, as soon as the parameter value is known, go back to the parameter section and enter the parameter value. To apply the value click **Update parameters**. The **Update parameters** button is only displayed if the parameter value has been entered.
   - In any step of a runbook, you can select the parameter section of the runbook in the **Details** on the right of the window. You might need to first expand the section if it is collapsed. The parameter can then be changed to a new value. To apply the new value click **Update parameters** (the **Update parameters** button is only displayed if the parameter value has been changed). As soon as a parameter has been used by a Runbook step, it cannot be changed.
8. Click **Start runbook**.
9. Follow the instructions in the runbook and proceed to the next step by clicking **Next step**.
10. If the step contains a command, use the **Copy** button within a command to copy the command to your clipboard. Open a command-line window and paste the command.
11. If the step contains an action, click on the action and select **Run** to run the action. **Note**: If running the action is not appropriate in the particular situation, then you can also choose to proceed without selecting **Run**. If you select **Run**, the action output is displayed in a text box. If an action produces JSON output, the JSON document is automatically formatted for better readability. Select **More info** to see description, parameter values, and type information.

    If the type of action is "Client-side", then the label of the **Run** button is replaced with **Launch**. Clicking this button launches the URL from the Client-side action in a new browser tab. You can review the resulting URL in the action output area and launch it again by clicking the displayed URL hyperlink.
12. You can pause or cancel the runbook by clicking **Pause** or **Cancel**. Pause temporarily suspends the execution of the runbook. You can pick up later from where you left off. You can use **Pause** to handle long running actions, for example actions that take longer than 10 minutes. The actions that are in progress continue while you pause the runbook. If you cancel the runbook you are prompted to select a reason. This feedback is helpful to the author of the runbook.
13. After you completed all steps, click **Complete** at the end of the runbook. Provide a rating, a comment (both optional), and a final assessment of the runbook. You are taken back to the **Library** page.
14. The **Activities** tab provides the **Runbook activities** table. Here you can find all the runbooks that were executed with information such as Status, Start time, Version, Comments, and if the runbook is automated. Filters such as user, runbook status, runbook type and runbook name can be applied. For more information, see Activities.

Load and reset example runbooks

Load ready-to-go sample runbooks to explore what you can do with runbooks.

Warning: If you want to edit a sample runbook and adjust it to your requirements, or if you are considering publishing a modified sample runbook to use in production, it is recommended to first create a copy of the sample runbooks and then edit and publish the copy. Similarly, if the sample runbook uses an automated step, then create a copy of the automation action and use that copy within your new runbook. This is to avoid losing your work (or even disrupting production) if somebody deletes or reloads the sample runbooks.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Data and tool connections**.
3. Ensure that a connection of type "SSH" has been configured. If not, you can configure it now (with default values).
4. From the main navigation menu, click **Operate** > **Automations**.

5. Click **Runbooks**.

   The **Library** tab is displayed by default.

Load sample runbooks

Load all pre-defined examples. You can also edit the example runbooks and create new versions.

1. Click **Load sample runbooks** on the upper right of the window (beside the **Create runbook** button) and select **Load sample runbooks**.
2. The sample runbooks are shown in the **Library** page. The runbooks will show up for Operators if they are published. The samples are marked by the text "Example:".
3. Click **Preview** to view the runbook and see what it does.
4. Click **Run** to run your first runbook. For more information about how to run a runbook, see Run your first runbook.

Sample runbooks with special administrative use cases

Some of the following sample runbooks go beyond just illustrating the features of runbooks:

- **Example: Convert existing operational procedures into runbooks**: This runbook explains how to convert your existing operational procedures into runbooks. It covers file based procedures and web/wiki based procedures.
- **Example: Import runbooks from an existing runbook deployment**: Provides instructions on how to export runbooks from one deployment and import them to another deployment. It uses the RBA API for both operations.
- **Example: Reduce amount of data in runbook data stores** and its automated variant **Example: Automatically reduce amount of data in runbook data stores**: Offload runbook execution data into a secondary data store for auditing or long-term storage purposes. This will delete that data from the production data store to ensure less disc space usage and faster response times on the active system.

Reset sample runbooks

Follow these steps if you edited the examples and would like to reload the original version, or if you have upgraded to a new version and want to get the latest versions of the examples:

- Open the runbook **Library** page.
- Open the **Reload sample runbooks** menu.
- Select **Delete sample runbooks** and wait for the sample runbooks to be deleted.
- Open the Load sample runbooks menu.
- Select the **Load sample runbooks** menu item and wait for the sample runbooks to be loaded.

Convert a sample runbook to a regular runbook

To save a sample runbook as a regular runbook:

1. On the runbook that you want to convert, click the menu overflow icon (the three dots at the end of the row) and select **Copy runbook**.
2. Change the runbook name. You can optionally update the description.
3. Click **Save**.

   After saving, the runbook will no longer be tagged as an example.
4. If the sample runbook references an action, then you need to create a "deep copy" manually:
   - Navigate to the **Actions** table and select the referenced sample action.
   - Copy the sample action and modify its name so you can identify the copy.

- Edit the copy of the sample runbook, and replace the reference to the original sample action with a reference to your new copy of the sample action.

Delete sample runbooks

1. Click **Load sample runbooks** and select **Delete sample runbooks**.
2. Confirm that you want to delete the sample runbooks from the **Library** page.

Creating a runbook

Document your existing operational procedures using the runbook editor. Share the knowledge and experience that you documented in a runbook with your team members.

Runbooks are created for one of the following reasons:

- Operators document their daily tasks in documents. Runbooks allow you to document and store these procedures in an ordered manner. When a runbook is published the knowledge can be shared between all operators.
- As a special case, create reusable runbook assets to combine those sequences of steps that otherwise would have to be repeated over and over again, and add them to your runbook as nested runbooks.
- A new set of events has occurred and a couple of skilled operators work out a new procedure to solve the issue that caused these events.

Procedure

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Runbooks**.

   The **Library** tab is displayed by default.
4. Click **Create runbook**. The runbook editor opens. For more information about how to use the editor, see Edit a runbook.
5. Provide data to describe the runbook. Make sure that you use a **Name** for the runbook that describes what problem this runbook solves.

   Restriction: Avoid using commas in runbook names. Runbook names containing commas cannot be filtered in the **Library** or on the **Activities** page.
6. (Optional): Provide a **Description**. The description contains information about specific situations that this runbook solves. Operators use the description to decide if a runbook meets their requirements.
7. (Optional): Add **Tags**. These help to filter a large amount of runbooks, as well as group runbooks with the same content.
8. Start adding steps to the runbook. Click **Add manual step**, **Add nested runbook step**, or **Add automatic step**.
9. Click inside the editor and describe the first step of your procedure. Use **commands**, **parameters**, **actions**, or **GOTO** elements to distinguish between different types of elements within a runbook. Remember to always create a new step if you start to describe a new task that the operator can run.
10. When all steps have been documented, click **Save draft** > **Save draft & close**.
11. Test the runbook. Search for your runbook on the **Library** page. Click **Run** to test the execution of the new runbook. Modify the runbook if necessary.
12. Publish your runbook.

1. Click **Edit** to open the runbook editor.
2. Click **Publish**. The runbook is now available to all operators.

Note: Runbooks are limited to a size of 1 MB. This relates to runbook metadata and its content (including embedded images). Attempts to create or edit runbooks in a way that exceeds this limit will result in an error.

Editing a runbook

The runbook editor provides editing capabilities as well as runbook elements that help to document operational procedures. Runbook elements are designed to avoid mistakes.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Runbooks**.

   The **Library** tab is displayed by default.
4. Click **Create runbook** to create a new runbook, or scroll through the list of runbooks and click the menu overflow icon and select **Edit** to open an existing runbook in the editor.

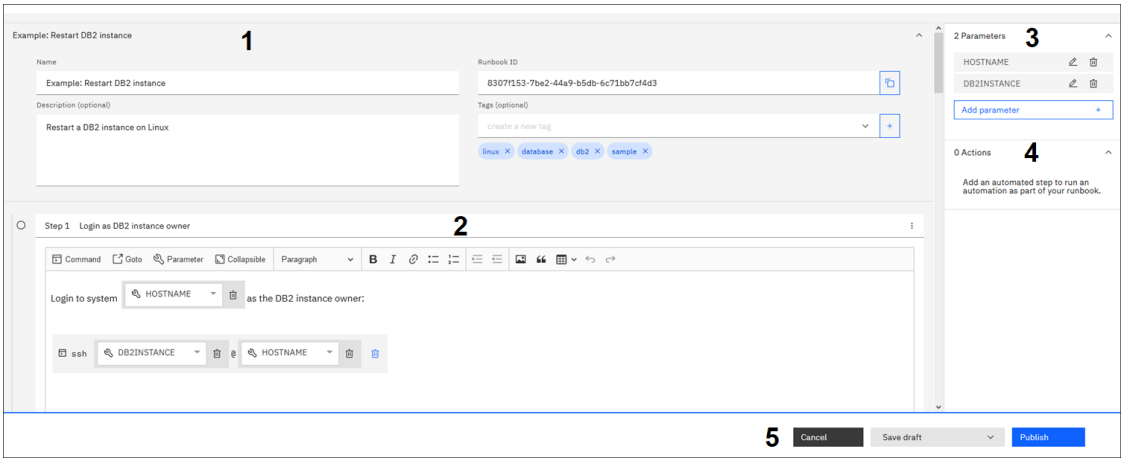


Figure. Runbook editor

Region 1 **Runbook details:** Enter descriptive data about the runbook.

- Name: the name identifies the runbook. Try to be as precise as possible and describe the problem that this runbook solves.
- Description: describe the problems that this runbook solves in more detail and sketch out the solution.
- Tags: provide tags to filter runbooks. You can select an existing tags from the drop-down list and click the plus icon "**+**" to add them. To create a new tag, type the text into the field, select the text string displayed, and click the plus icon "**+**".
- Runbook ID: this field is generated by Runbook Automation. The Runbook ID can be used to identify the runbook using the Runbook Automation HTTP REST Interface. Use the copy button to copy the ID to the clipboard.

Region 2 **Editor:** Provides text edit actions and runbook elements to describe operational tasks. Each step has its own section with an optional title. Manual steps cannot contain automation elements. Automated steps must be added for actions.

Region 3 **Parameters:** Create parameters that are used in this runbook.

Region 4 **Actions:** Add pre-defined actions to the runbook. Action steps only contain an action, no text.

Region 5 **Save and Publish** the runbook.

For more information about editing a runbook, see the following topics:

- Runbook elements and editing actions
- Adding actions
- Adding parameters
- Parameter types
- Adding a GOTO element

Runbook elements and editing actions

Use the editor to describe your daily tasks step by step. The editor provides editing actions and runbook specific elements to help you write the instructions.

Click **Add manual step** and use the following runbook elements and text editing actions to edit and format your operator instructions.

Icon	Name	Editing action
Command	Add command	Click the editor canvas. Select **Add command** and enter your command. Click **Enter** to quit the command and continue in the next row. Another way to enter a command is to mark the command and click **Add command**. As the command is still selected, you have to click on the canvas to deselect the command. Click **Enter** to quit the command and continue in the next row. **Keyboard shortcut:** Use the **tab key** to move the focus on the canvas. Press Ctrl + 4 and begin typing your command. **Remove command:** You can remove the command by placing the cursor in the command and clicking **Remove**.
Goto	Add GOTO	Click the editor canvas. Select **Add GOTO**. A goto element is added with the label **END**. Click the GOTO element and select the step that you want to jump to. You can select any subsequent step. **Keyboard shortcut:** Use the **tab key** to move the focus on the canvas. Press Ctrl + 5. The GOTO element is added. **Remove GOTO:** Select the GOTO element and click **Remove**. For more information about how to use the GOTO element, see Adding a GOTO element.
Parameter	Add Parameter	To add a parameter, go to the **Parameter** palette and drag the parameter that you require to the canvas. Or mark the text which should be a parameter name and click **Add parameter**. **Keyboard shortcut:** Use the **tab key** to move the focus on the canvas. Press Ctrl + 8. A parameter menu is added. Click the arrow and select the required parameter. **Remove parameter:** Select the parameter and click **Remove**. For more information about creating parameters, see Adding parameters.
Collapsible	Add collapsible details	To add collapsible details, click the icon and enter a title for the collapsible section. Then enter the information that you want to collapse and expand in a section. **Keyboard shortcut:** None. **Remove collapsible details:** Use the delete or backspace key to delete collapsible details.
Paragraph	Paragraph or Heading	Switch between paragraph style or select from the available heading styles.
B	Bold	Select a text and make it bold by selecting **b**.
I	Italic	Select a text and make it italic by selecting **i**.
	Insert Link	If you want to insert a link, click **Insert Link** and enter in the URL field the link. Do not forget to provide the link text and decide whether you want this link to be opened in another tab.
	Bulleted list	Select text that is spread over a couple of rows and click bulleted list. All lines are indented and bulleted.
	Numbered list	Select text that is spread over a couple of rows and click numbered list. All lines are indented and numbered.
	Increase indent	Select text that is spread over a couple of rows and click increase indent. The whole text section is moved to the right.

Icon	Name	Editing action
	Decrease indent	Select text that is spread over a couple of rows and click decrease indent. The whole text section is moved to the left.
	Insert image	If you want to add a screen capture for example, select **Insert Image**.
	Block quote	If you want to add a quotation that is set off from the main text, select **Block quote**.
	Insert table	If you want to insert a table, click **Insert Table**. Enter the number of rows and columns that you need and specify the column with and width unit.
	Undo	You can undo your last 20 editing actions.
	Redo	You can redo your last 20 editing actions.

Adding actions

Actions are used to run several steps automatically. Use the **Actions** page to create actions.

The job of an operator can involve repetitive tasks. For example, *"logon to DB2"* or *"start server dagobert45"*. Actions are scripts or Ansible templates that document these steps. If you use actions within your runbook then the operator does not have to manually run these steps. They can run all steps with one click.

For more information about how to create actions, see Creating an action.

The runbook editor displays any created actions in the **Actions** pane.

Manual steps cannot contain action elements. Automated steps must be added for actions.

It is common practice for actions to contain parameters. When adding actions to a runbook you must decide how the values of the action parameters are filled by creating a parameter mapping. Select from the following options:

- **Use a runbook parameter:** a runbook parameter, which can be filled manually by an operator executing the runbook, is used to fill the specific action parameter. When adding an action to a runbook, you can select existing runbook parameters or directly create a new runbook parameter.
- **Define a fixed value:** the action added to the step of a runbook will always be launched with a fixed value for the parameter. (If the same action is used in a different runbook – another fixed value could be defined there.)
- **Use the default from the action:** if the action defines a default value for this parameter, it is possible to use that value.
- **Use the output of a previous action:** the output of an action that was added to a previous step can be used as input for the action of the current step.
- **Use the logged in user:** the parameter value will be filled with the username of the user, who is logged in to Runbook action at the time when the runbook is run.

Note: If the username contains the @ symbol, for example because the username is an email address, the @ symbol and all characters that follow it are removed from the username.

Procedure

1. In the New (or edit) runbook editor, click **Add automated step**.
2. Search for the action. Click the search icon and type in the search term that describes the action that you need.
3. Choose the action that you want and then click **Select this action**.
4. Select the parameter mapping type. A window opens with action details and the parameter definitions. In the **Mapping** column, select how action parameters are to be filled with values. You can choose from the following options:
    - **New runbook parameter:** add a parameter. If default values are provided, update the default values as required. Enter a new parameter name and description.
    - **Fixed value:** an entry field is provided to enter the parameter value.

- **(Existing) runbook parameter:** select the parameter from the list of existing parameters.
- **Use default:** use the default value set by the action.
- **Action output:** use the action output from a previous runbook step. The output value of the action will be used as the parameter value for the current action. Note, if the action in the previous runbook step is a script action or a HTTP action then you can filter the output of that action as described in Creating SSH actions or Creating HTTP actions, respectively. This option is only available if the runbook contains a previous step with an action.
- **Use logged in user:** select this option to fill the parameter value with the username of the user who is logged in to Runbook action at the time when the runbook is run.

Note: If the username contains the @ symbol, for example because the username is an email address, the @ symbol and all characters that follow it will be removed from the username.

Next steps

- **Edit parameter configuration:** if you want to edit the parameter configuration of your action, select the action and click **Change parameter mappings**. Change your settings and click **Save**.
- **Action with errors:** if the action is decorated with an error symbol, then the parameter settings are not correct. Click **Change parameter mappings** to correct the parameter settings.
- **Delete an action:** if you want to remove the action, click **Remove selection**.

Adding nested runbooks

Nested runbooks are used to collect sets of steps that are commonly used, so they can be reused by other runbooks.

About this task

Create reusable runbook assets to combine those steps that otherwise would have to be repeated over and over again in various runbooks. For example, if you have a 3-step login procedure, you can describe this procedure once in a runbook, and then reuse this runbook by adding it as a "nested runbook step" within all other runbooks that require the 3-step login procedure. When the 3-step login procedure needs to be changed to become a 5-step login procedure, you only need to change the nested runbook. Once the nested runbook has been published, all runbooks that refer to this nested runbook inherit the updated 5-step procedure.

Once the reusable runbook has been created and published, it is ready to be referenced by other runbooks as a nested runbook.

Runbooks that include nested runbooks are always considered as "manual" runbooks, even if all nested runbooks contain automated steps only, and the runbook itself does not contain manual steps.

Nested runbooks allow a maximum nesting depth of 10 runbooks, and the complete nesting structure cannot contain more than 100 nested runbooks. Attempts to create or change runbooks in ways that exceed these limits will fail.

The statistics for nested runbooks are unchanged when running the parent runbook. Statistics (such as success and failure, but also rating and feedback comment) are always recorded at the level of the runbook that was started, and not propagated to any nested runbooks.

It is common practice for nested runbooks to contain parameters. When adding a nested runbook to a runbook you must decide how the nested runbook's parameter values are filled by creating a parameter mapping. Select from the following options:

- **Use a runbook parameter**: a runbook parameter, which can be pre-filled automatically from an event or edited manually by an operator executing the runbook, is used to fill the parameter of the nested runbook. When adding a nested runbook to a runbook, you can select from existing runbook parameters or directly create a new runbook parameter within the parent runbook.

- **Define a fixed value**: the nested runbook added to the step of a runbook will always be launched with a fixed value for the parameter. (If the same nested runbook is used in a different runbook, another fixed value could be defined there.)
- **Use the default value from the nested runbook**: if the nested runbook defines a default value for this parameter, it is possible to use that value.
- **Use the output of a previous automation**: the output of an automation that was added to a previous step can be used as input for the nested runbook of the current step.
- **Use the logged in user**: the parameter value will be filled with the username of the user who is logged in at the time when the runbook is run.

Note: If the username contains the @ symbol, for example because the username is an email address, the @ symbol and all characters that follow it are removed from the username.

Procedure

1. In the New (or edit) runbook editor, click **Add nested runbook step**.
2. Search for the published runbook that you want to add. Click the search icon and type in the search term that describes the runbook that you require as a nested runbook.
3. Choose the runbook and click **Select this runbook**.
4. If the selected runbook contains parameters, you are now asked to define the parameter mappings. A window is displayed with the list of parameters that are defined in the nested runbook. In the **Mapping** column, select how each of the parameters of the nested runbook is to be filled with its corresponding value. You can choose from the mapping options that are described above.

Next steps

- **Edit parameter configuration**: if you want to edit the parameter mapping, click **Change parameter mappings**. Change your settings and click **Apply**.
- **Nested runbook with errors**: an error message is displayed when you open the runbook editor again after the parameter list of the nested runbook has been changed, so the parameter mappings are no longer valid. This can happen when the parameter list of the nested runbook has been changed, and the changed nested runbook has been published again. Click **Change parameter mappings** to adjust the parameter mappings.
- **Delete nested runbook**: if you want to remove the nested runbook, click **Remove selection**.

Adding parameters

Parameters are used as general placeholders for values that are used on a frequent basis. Examples of parameters are **DATABASE_NAME**, **DIRECTORY**, **HOSTNAME**, **URL** and so on.

You can use runbook parameters in the following scenarios:

- Parameters can be used as variables that get substituted by a value in the text of a runbook step, for example:

```
STEP 2
In this step you determine the longest running process of system $SYSTEM
```

- As input to a command defined in a runbook, for example:

```
STEP 5
Now issue the command: cd $DIRECTORY
```

- Parameters of an automation can be filled with the value of a parameter defined for the runbook.
- Actions of type Script have a system parameter `target` which defines the target system where the script is run.

- Actions of type Script have a system parameter **`user`** that specifies the UNIX username which is used to run the script on the target system. This parameter can be mapped automatically to the user who is currently logged in to Runbook Automation. For more information, see Adding actions.

Operators enter values such as **`username`**, **`password`**, **`hostname`**, **`database name`** or **`URL`** on a frequent basis. If you have to deal with many different computer systems, you must remember numerous user names, passwords, host names or database names. In order to avoid typos or having to remember values, you can define parameters. For example, you can define a parameter **`DATABASE_NAME`** and either provide a default value or enter the value when you run the runbook. Default values are the best fit if you do not want operators to remember and enter the value.

Parameters are local to the runbook. They are not available for other runbooks.

Parameters are used within runbooks and for actions. Within runbooks, you can use parameters either standalone, for example **`URL`**, or within a command, for example **`connect to HOSTNAME`**.

Parameters must be filled with values when executing a runbook. This can be done in the following ways:

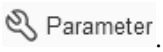
- When a runbook is launched by clicking the execute button from the Runbook Automation UI, an operator must first enter the parameter values. The operator is prompted with the name and description of all parameters defined for the runbook, as well as an optional default value.
- When a runbook is launched from an event, a Trigger defines which values of the event data are mapped to which parameter of the runbook. Values that are entered in this way cannot be changed.

Runbook parameter values can be changed during the execution of a runbook. In any step of a runbook, you can select the parameter section of the runbook in the sidebar on the left. You might need to expand the section first, if it is collapsed. The parameter can then be changed to a new value. To apply the new value click **Update** (the Update button is only displayed if the parameter value has been changed). Steps of the runbook that are not yet executed will reflect the new value.

Runbook parameters can be used to define parameter values for actions within the same runbook. Parameter values for actions can be filled using the result of a previously executed automation.

In this way, the parameter Action output of a previous automation **`Find large file systems`** is an automatic parameter available for actions in the current step. It is possible to use the output of any automation that ran before in this runbook.

Procedure

1. Create a parameter. Click **Add Parameter** 

. A dialog box opens where you can enter the parameter name, the description, and the default value. If you select **Optional** you can start the runbook execution without defining a value, and set the parameter value during runbook execution. This setting is useful if the parameter value is not available at the beginning of the execution, but during execution.

   When you enter the runbook parameter name you can use alphanumeric characters, characters from national languages, and some special characters. But do not use the special characters ampersand (&), backslash (\), space ( ), greater than (>) or less than (<) when you define runbook parameter names.

   As a result, a parameter of the runbook is created and can be used as a variable in the text of a step, as a variable value of a command, or as input of an automation added to a runbook step. For more information about how to use parameters, see Runbook elements and editing actions.

2. Edit a parameter. Click **Edit** next to the parameter that you want to edit.
3. To edit a parameter used by an automation, click **Edit** for the automation you have added to a runbook step.

Parameter types

Runbook Automation supports multiple parameter types. You can select parameter type when you create or modify a runbook parameter in the runbook editor, or when you create or modify an action parameter in the edit action dialog.

Depending on the action type, certain parameter limitations might apply. Supported parameter types are string, integer, Boolean, and array.

String

When defining a parameter of type string, an operator running your runbook has a free text input element to enter a parameter value. You can specify the minimum or maximum number of characters that the operator can use. You can also mark a string parameter as a password so that any input that is entered by the operator is not displayed.

Any runbook or action parameter that is created before June 2021 is of type string.

Integer

When defining a parameter of type integer, an operator running your runbook has an integer input element to enter a parameter value.

You can specify the minimum or maximum value for the integer value.

Boolean

When defining a parameter of type Boolean, an operator running your runbook has a toggle element to enter a parameter value.

Array

When defining a parameter of type array, an operator running your runbook has an input element to define a various number of inputs. The array parameter must define what type of inputs are allowed. You can use an array of either strings, integers, or Booleans.

Enumerations

To restrict the input for a specific parameter to a discrete set of values, you can extend the parameters with an optional enumeration list. Enumeration support is available for parameters of the following types:

- string
- integer
- arrays with string or integer as their item type

You can define any number of allowed discrete values for your enumeration. The user will then be given the choice of selecting any one of them in the Runbook Automation UI, or multiples of those in the case of an array parameter. If a user tries to send a value which is not part of the enumeration, for example through an API call, the action is rejected with an error message.

Additionally, you can assign labels for any or all of the values of an enumeration for integer or string parameters. Labels are optional. If a label is present, the label is displayed in the Runbook Automation UI together with the value.

Example: If you are writing an application which accepts the days of the week as numeric values from 0-6, you can define the following enumeration list with labels:

Value	Label
0	Monday
1	Tuesday
2	Wednesday
3	Thursday
4	Friday
5	Saturday
6	Sunday

This way, the user can only choose a valid value (0-6), but is also shown the appropriate meaning of the value, as presented by the label.

Action parameter support

The following table contains details about the parameter type to use for each type of automation.

Action type	String	Integer	Boolean	Array
Script/Bash	String value is exported as a variable.	Integer value is exported as a variable.	Boolean is exported as variable with either value "true" or value "false".	Not supported
Script/Powershell	String value is available as a variable of type `System.String`.	Integer value is available as a variable of type `System.Int32`.	Boolean value is available as a variable of type `System.Boolean`.	Array variable is available as a variable of type `System.Array`.
HTTP	String value is substituted where ever the parameter name occurs in HTTP action API endpoint, credentials, or body.	Integer value is substituted where ever the parameter name occurs in HTTP action API endpoint, credentials, or body.	Not supported	Not supported
Ansible	Ansible parameters extra variables, limit, tags, skip tags, job type and all survey parameters of type text, password, or multiple choice are harvested as string parameters.	Ansible parameters verbosity, inventory, credentials, and all survey parameters of type integer are harvested as integer parameter types.	Ansible parameter diff mode is harvested as Boolean parameter type.	Survey parameters of type "multiselect" are harvested as an array parameter type.

Adding a GOTO element

The GOTO element is used to skip steps or jump to the end of the runbook.

See the following example scenarios where the GOTO element is used.

- **Example: Run optional steps as result of a check**

  1. Log in to the server where you want to start the application server.
  2. Check the current CPU consumption. If it is less than 80% **GOTO Step 5**.
  3. Find the process consuming the most CPU.
  4. Determine the process.
  5. Start the application server.
  6. Check the log file for confirmation that the application server is started.
- **Example: The execution was successful and you can complete the runbook**

  1. Restart the Application Server.
  2. If this was successful, then GOTO END.
  3. Check your log files.

Configuring runbook creation

Use these procedures to define the runbook creation process and control the publication of runbooks.

- Configuring runbook approval

Define whether to use the approval workflow or the direct publish workflow to make runbooks available for execution by operators.

- Runbook approval process

  The Runbook approval process offers control over the publication of runbooks. This is beneficial if a mandatory review process is necessary, if prematurely released runbooks could be harmful, or when auditable information on the publication is required.

Configuring runbook approval

Define whether to use the approval workflow or the direct publish workflow to make runbooks available for execution by operators.

1. Navigate to the **Runbooks Library**.
2. Click **Settings** next to the **Create runbook** button.
3. Select **Approve runbooks before publish** if you want runbooks to be approved before they are published.
4. If you want to enforce that an approval must be refreshed at a regular interval, select **Set expiration date after xxx days** and choose a time interval. If this option is enabled, the runbook approval will expire after the specified number of days and can no longer be executed until the runbook is reviewed, submitted for approval, and approved again.

   The deactivated approval process is described in Create a runbook.

   The activated approval process is described in Runbook approval process.

   Note: If the approval process is deactivated, all pending approvals are deleted.

Runbook approval process

The runbook approval process offers control over the publication of runbooks. This is beneficial if a mandatory review process is necessary, if prematurely released runbooks could be harmful, or when auditable information on the publication is required.

Figure 1. Runbook approval process

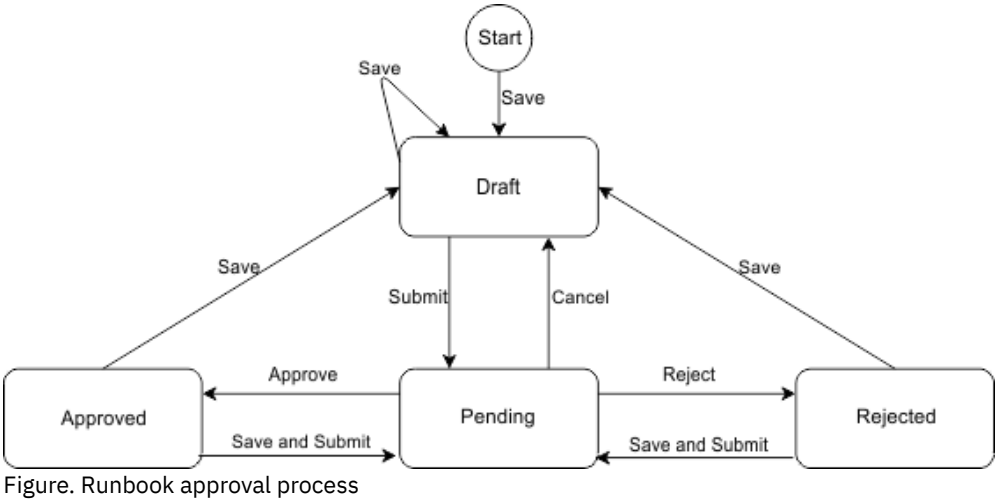


Figure. Runbook approval process

Figure 1 illustrates the states and transitions of the runbook approval process. If the approval process is enabled, it is not possible to directly publish a runbook. Instead an option to submit the runbook is displayed. The following options are then available:

List all runbooks that must be approved

1. Open the **Library** page.
2. From the **Status** filter drop-down menu, select **Approval pending**.
3. All runbooks for which approval is pending are shown.

List all runbooks that are approved

1. Open the **Library** page.
2. From the **Status** filter drop-down menu, select **Approved**.
3. All approved runbooks are shown.

Note: You can also filter for a status of **Approval expired** and **Approval rejected**.

Submit a runbook draft for approval

1. Create or edit a runbook and click **Submit for approval**.
2. Enter the name of the approver in the field provided and click **Send**.

The runbook is now in a pending state. It can no longer be edited, except for changes to its state.

Cancel a pending runbook approval

1. Locate the runbook on the **Library** page.
2. Click the menu overflow icon (the three dots at the end of the row) and select **Remove approval**.

The runbook is now in normal draft mode and can be edited. The assignee can no longer approve or reject the runbook.

Approve a runbook

Only the assigned person from the submission can approve a runbook.

1. Locate the runbook on the **Library** page.
2. Click the menu overflow icon and select **Approve**.

The runbook now has a status of approved, which is the default for execution. New drafts can be created. Information pertaining to the approval assignee and timestamp is stored within the runbook.

Reject a runbook

Only the assigned person from the submission can reject a runbook.

1. Locate the runbook on the **Library** page.
2. Click the menu overflow icon and select **Reject**.

The runbook now has a status of rejected and can be edited again. Information pertaining to the approval assignee and timestamp is stored within the runbook until a new draft version is created and the old information is deleted.

Runbook versions

You can use versioning to create incremental improvements to runbooks and monitor the success of changes. Subject Matter Experts can continuously improve the quality of runbooks by using the comments provided and the available run metrics. As a result, different versions of a runbook are created.

All available runbooks are listed on the **Library** page. Click the menu overflow icon (the three dots at the end of the row) and then click **Preview** > **Versions** to display all the versions of a runbook.

- **Draft**

  The Draft version is a work-in-progress runbook. It is marked with a draft flag. A draft runbook is not yet published and does not run in production. It is not visible to any operator. You can run and test the current draft.

- **Latest published**

  The latest published version is the version that is running and used by all operators. This is the version that the operations analyst can preview and work with.

- **Archived**

  Archived versions are previously used versions. They are not in production and cannot be used by any operations analyst. The metrics of the previous versions help the Subject Matter Expert to monitor continuous improvements of the runbook. You can run archived version to learn why this runbook did not run successfully.

Save and publish a runbook

Creating a runbook involves documenting complex steps. These steps must be reviewed and tested before you can publish the runbook for use in production.

After a runbook is published and used in production, you might improve it based on the feedback that you receive from comments, success rate, and ratings. The runbook editor provides the following actions.

- **Cancel:** closes the runbook editor without saving any changes.
- **Save as:** use this action to save a copy of the runbook using a different runbook name.
- **Save draft:** saves the updates that are added. The editor remains open. **Save draft** is useful if you are editing a runbook and you want to take a break and continue later. Or you just want to save your changes as you go.
- **Save draft & close:** saves your changes and closes the editor. You use this option if you need to stop and work on something else.
- **Publish:** this button is only available when the approval process has been deactivated. After a runbook has been published it can be used by all operators in a production environment.
- **Submit for Approval:** this button is only available when the approval process is enabled.

If the approval process has been activated, see Runbook approval process for more information about approval before publication.

Filtering runbooks

Filter by status, groups or tags to easily locate runbooks.

1. Open the **Runbooks Library** page.
2. To filter for runbook status, select one or more items in the **Status** filter. The filter matches any selected status (or condition).

   Note: Filtering for status **Draft** displays runbooks that have never been published. Draft versions of already published runbooks are not displayed.

- To filter for tags, select one or more items in the **Tag** filter. All tags must match to apply the filter (and condition).
- To filter for authors, select one or more authors in the **Authors** filter. The filter matches any selected Authors.
- If a combination of several filters are used, all filters must match to apply the filter (and condition).
- Additionally, you can filter by runbook name using the **Search** field.

  The search field allows to search using regular expressions. If you want to search for special characters that are used by regular expressions, for example plus (+), asterisk (*), question mark (?), brackets (() [] {}), or backslash (\), you must escape those characters. For example, if you want to search for (T), you must enter \(T\) as the search string.

  Note: The filter drop-down list displays up to 50 available filtering options. You can start typing the name of the filter option that you are looking for. As you type, the list displays only items that contain the typed text.

Assigning runbooks to groups

Each runbook can be assigned to one or more groups, or left unassigned.

Runbooks can only be assigned to user groups if the respective feature is by an administrator. If the feature is enabled and if you only want to check what groups a runbook is assigned to, complete steps 1 - 3 from the instructions below for "Using the feature" but do not make any changes.

Assigning runbooks to groups allows you to organize runbooks in a more efficient manner. There are two main reasons for assigning runbooks to groups:

- Groups determine the visibility of runbooks for operations analysts.
- It is easier to find relevant runbooks in the **Library** page when Filtering runbooks by groups.

Part 1: Enabling the feature

1. Navigate to the Runbook **Library** page.
2. Click the Configure settings icon next to the New runbook button.
3. Select the **Allow users to assign runbooks to groups** checkbox.
4. Save the changed setting.

Part 2: Using the feature

1. Go to **Library**.
2. Use the check boxes to select one or more runbooks.
3. An action bar is displayed. The bar shows available actions and how many runbooks are selected.
4. On the action bar, click **Grant permission**.
5. Use check boxes to choose which groups the selected runbooks should belong to. You can also select all or clear all groups by using the checkbox in the header row.

   Note: Some checkboxes might appear selected, but dimmed. This is an intermediate state that occurs when some (but not all) of the selected runbooks belong to the group.
6. Click **Assign** to confirm the selection or **Close** to discard any changes.

Export and import runbooks

Export and import runbooks by sending requests to the HTTP API. You can use existing tools to send requests, for example cURL, Postman for Chrome, or RESTClient for Firefox.

Before you can send requests, you need to create an API key for authentication. See API Keys and navigate to section "Get authentication by using Zen API Key" for instructions how to create and encode an API key ("ZenApiKey").

In addition, determine the host name part of your IBM Cloud Pak for AIOps user interface URL. The base URL for sending API requests has the format **`https://<waiops-hostname>`**, for example, **`https://cpd-cp4waiops.apps.mycluster.mycompany.com`**.

Runbooks are identified by an ID that gets generated when you create the runbook. For example, this ID is used when you create a runbook policy to link the runbook to alerts that match a specific filter pattern. When you plan to export the runbook and import it again at a later time (for example, to restore a backup version of the runbook after the runbook has been removed), you must use the "**`keepId`**" parameter to include the runbook ID during the export. This is so the integrity of the runbook policy is preserved when the runbook is imported later on.

Depending on your certificate configuration, you might need to add the "**`-k`**" or the "**`--insecure`**" option for the sample curl commands listed below.

Export Runbooks

The following two calls create output, which can be imported again.

Use the following API to export all runbooks (including runbook IDs):

```
GET api/v1/rba/runbooks?exportFormat=keepId
```

When enabling the option exportFormat via the query parameter, the usual output of the command is replaced by an array of JSON objects. These JSON objects can be used to create the runbooks by using the import API endpoint. The following call is an example:

```
curl --header "Authorization: ZenApiKey <ZenApiKey>" https://cpd-cp4waiops.apps.mycluster.mycompany.com/aiops/api/story-manager/rba/v1/runbooks?exportFormat=keepId
```

Replace **`<ZenApiKey>`** with the encoded Platform UI (Zen) API key token that you created before.

Use the following API to export a single runbook:

```
GET api/v1/rba/runbooks/<RunbookId>?exportFormat=keepId
```

Replace **`<RunbookId>`** with the ID of the runbook. You can retrieve this ID, for example, from the **Edit Runbook** window. The following call is an example:

```
curl --header "Authorization: ZenApiKey <ZenApiKey>" https://cpd-cp4waiops.apps.mycluster.mycompany.com/aiops/api/story-manager/rba/v1/runbooks/*<RunbookId\>*?exportFormat=keepId > exportedRunbook.json
```

Replace **`<ZenApiKey>`** with the encoded Zen API key token that you created before. Replace **`<RunbookId>`** with the ID of the runbook.

Import Runbooks

Use the following API to import runbooks:

```
POST api/v1/rba/runbooks/import
```

The following call is an example:

```
curl -H "Expect:" -H "Content-Type: application/json" --header "Authorization: ZenApiKey <ZenApiKey>" -d@./exportedRunbooks.json https://cpd-cp4waiops.apps.mycluster.mycompany.com/aiops/api/story-manager/rba/v1/runbooks/import
```

Replace **`<ZenApiKey>`** with the encoded Zen API key token that you created before. Note: **`./exportedRunbooks.json`** is the output file that is created from the previous example.

Note: If you previously exported a lot of runbooks (or some very large runbooks), you might face the 10 MB upload limit of the Runbook API when trying to import them again. The runbook file must be split into multiple JSON files, and each of these chunks must be imported separately. The following examples assume that you have the **jq** tool installed.

- **Example**: Split by maximum file size

  If your runbooks do not reference any automation actions, or if your automation actions are not shared between multiple runbooks, you can use the following command to split the file into chunks of a maximum of 10 MB each, named **exportedRunbooks.json.aa**, **exportedRunbooks.json.ab**, etc.:

  ```
  cat exportedRunbooks.json | jq .[] -c | split -C 10485760 --filter='jq . -cs > $FILE' - exportedRunbooks.json.
  ```

- **Example**: Split by automation actions

  If **exportedRunbooks.json** is larger than 10 MB and if it contains multiple runbooks that reference the same automation action, then try to group all these runbooks into a single chunk file. This avoids creating multiple copies of the same automation action during the import. You can use the following command to split the original file into multiple JSON chunk files, named **exportedRunbooks.json.aa**, **exportedRunbooks.json.ab**, etc. Each of these files contains all the runbooks that reference the same set of automation actions:

  ```
  cat exportedRunbooks.json | jq 'group_by(.automations)'[] -c | split -l 1 - exportedRunbooks.json.
  ```

General Notes on export and import

- Exporting runbooks returns the data of referenced automations inline. These automations are created new when the runbook is imported.
- Import actions do not overwrite existing runbooks, automations, or both.
- Exporting multiple runbooks at once will consolidate automations that are referenced by more than one runbook and only create the minimum necessary number of new automations.
- Imported runbooks are in draft state and do not retain any kind of version, execution, or approval history. The user performing the import action is visible as the runbook creator.
- Exporting runbooks returns the data of referenced runbooks depending on the chosen export format mode.
- In normal mode, nested runbooks are exported in the "flat" structure, where all runbook references are resolved and runbooks are exported as if they did not contain references to other runbooks.
- In keepId export mode, nested runbooks are exported while retaining their referencing structure. Referenced runbooks are included in the data as additional runbooks. Runbooks that are referenced by their latest version keep their ID. Runbooks that are referenced by a fixed version create an additional entry with a placeholder ID, which is replaced by a new random ID upon importing the data set.
- Importing a data set which includes nested runbooks fails to create any runbooks referencing other runbooks if the "publish" flag is not set. This occurs because referencing other runbooks is only allowed on published runbooks, not on drafts.
- When exporting multiple runbooks, ones with invalid nested runbook references and invalid automation action references will not be exported. The export will include all other runbooks, and all errors are logged in the runbook service.

For more information about creating an API key, see API Keys.

For more information about the API, see Automation (runbook) API.

Related links

Installing new SSH key during migration

The SSH key that is associated with an SSH connection cannot be moved or copied to another deployment.

Delete a runbook

Users assigned the Service Administrator, Automation Administrator, or Administrator role can delete a runbook.

1. Open the **Runbooks Library** page.
2. Use the check boxes to select one or more runbooks.
3. An action bar is displayed showing the available actions and the number of runbooks selected.
4. On the action bar, click **Delete** . If you delete a runbook, all versions and the activity history of the runbook are deleted as well.

Activities

See recently executed runbooks, including runbooks in progress. Up to 1000 records matching the current criteria are displayed on the **Activities** UI. More records can be retrieved through the runbook API. If you need to see the records on the UI, use the filters on the page to refine the search criteria so that less records match.

On the **Activities** page, you can find all runbooks that have been started. You can filter your list by the following criteria:

- User

  Click the **User** filter and select one or more user names. All runbook activities for the selected user names are shown.

- Runbook name

  Click the **Runbook** filter and select one or more runbook names. All runbook activities for the selected runbook names are shown.

- Runbook type

  Click the filter **Type** and select **Automatic Manual**. Fully automated runbooks are runbooks in which each step contains an automation. An operator does not need to interact with the runbook. Automated runbooks can be mapped with events by creating a trigger. To see the automated runbooks that are started by a trigger, click **Automatic**.

- Runbook execution status

  Click the **Status** filter and choose a status such as Success, Failed, Cancelled, In progress, or Completed. For example, you can find all runbook activities that are paused by selecting **In progress**. When you are ready to return to the task you can resume the runbook. Select the **Canceled** status if you want to see all canceled runbooks and the reason why they have been canceled.

Monitor runbook history

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Runbooks**.
4. Select a runbook.
5. Click the menu overflow icon (the three dots at the end of the row) and select **View history**.

   The activity history table and runbook details such as description, type, rating, success rate, and activity statistics are displayed.

6. Use the filter boxes on the page to display only specific runbooks, for example:

   1. To see runbook activities for a specified time frame, select the dates in the **Start** and **End** fields.
   2. To see runbook activities that are still in progress, select **In progress** from the Status filter. If you want to resume a runbook execution, click **Resume**.

Actions

In runbooks, actions are the collection of several manual steps into a single automated entity. An action improves runbook efficiency by automatically performing procedures and operations.

The Service Delivery Engineer can provide an action to replace frequently used steps with a single click. Depending on the action type, defining a connection might be a prerequisite. For more information about how to set up a connection, see Configuring data and tool connections.

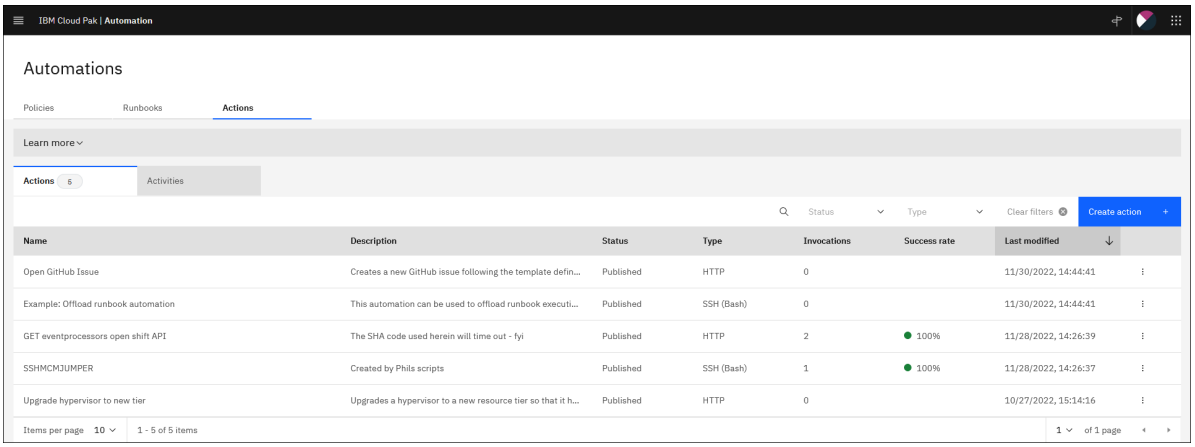


Figure. Actions UI

For more information about runbook actions, see the following topics:

- Managing actions
- Creating an action
- Ansible actions
- Predefined actions
- Action versions
- Save and publish an action
- Testing an action
- Running an action
- Activities

Managing actions

As a Service Delivery Engineer, you are interested in how the actions run and how to keep them running smoothly. Use the **Actions** page to test actions, view statistics, and administer the available actions.

The following types of runbook contain actions:

- Semi-automated runbooks: runbooks that contain one or more actions. These runbooks consist of manual and automated steps.
- Fully automated runbooks: runbooks that contain an action in each step. If fully automated runbooks are started by a trigger, you can find their status on the **Activities** page. Select the filter **Type** and select **Triggered automatically**.

On the **Actions** page, you can find all available actions with the following information displayed in the table:

- **Name**

  Name of the action.

- **Description**

  Description of the action.

- **Type**

  Actions are of the type SSH, HTTP, Client-side, Predefined, or Ansible, depending on the connection type used.

- **Invocations**

  The number of times the action has been used in runbooks.

- **Rating**

  Average rating, based on the ratings provided by operators and administrators who ran the action and provided a rating.

- **Success rate**

  How reliably did this action run so far? The success rate indicates how well an action ran.

- **Last modified**

  A timestamp of when the action was last modified.

- **Actions**

  As a Subject Matter Expert you can **Test**, **Edit**, **Copy**, and **Delete** an action unless the action is of type **Predefined** or **Ansible**.

  Select an action in the table to display the **Preview** side panel for that action. You can Edit, Copy, Test, and Delete actions directly from the side panel. The **Details** section is open by default and displays the **Description**, **Action ID**, **Prerequisites**, **Success rate**, and **Last modified by**.

  The **Parameters** tab contains the name, description, and default values of the parameters contained in the action.

  The **Content** tab of the side panel shows the script content in the case of script actions, or the Method and API endpoint in the case of HTTP actions.

  Click the **X** in the top right corner to close the side panel.

How is the success rate of an action calculated?

An action can have the following exit states:

- **failed** - The action could not even be started, for example because of an invalid hostname, unsuccessful authentication, or incorrect parameters (in case of an Ansible action).
- **unsuccessful** - The action could be started but has a return code indicating an error, for example for ssh actions the exit code is not 0, for http actions the http return code is >= 400.
- **successful** - The action could be started and has a return code indicating success, for example for ssh actions the exit code is 0, for http actions the http return code is <400.

The success rate indicates the quality of an action. For example, for ssh actions it indicates the quality of the corresponding scripts. For Ansible actions, the success rate indicates the quality of the corresponding Ansible playbooks.

It does not indicate if the configuration is correct. For example, if an ssh action cannot be executed because of an invalid hostname or authentication failure, you will see this in the success rate of the runbook, but not in the success rate of the action.

Therefore, the success rate is calculated using the number of unsuccessful and successful action activities. It does not include the number of failed action activities.

The total number of action invocations that is listed in the **Actions** table includes all action activities, including the failed ones.

Creating an action

Create an action by defining the action type and configuring the parameters and fields.

Actions are routines that run a task automatically. There are different types of actions depending on how tasks are run. Runbook Automation supports actions of type SSH (script), HTTP, Client-side, Predefined, and Ansible. For actions of type script and Ansible, a connection must be configured. Ansible automations are harvested directly from Ansible, so they appear in the automations table as soon as the Ansible connection is configured. For more information about how to set up a connection, see Configuring data and tool connections.

- **Creating SSH actions** - You can create an action of type script by using the **SSH provider**. The Operating System of the endpoint system can be either UNIX® (including AIX® and Linux®) or Windows™. For UNIX®, the SSH provider executes a script using the default shell. Alternatively, a shebang (#!) can be used to execute the script in any scripting language the target system can interpret. The SSH provider does not require an additional agent as it uses a direct remote execution connection through SSH.
- **Creating HTTP actions** - The HTTP action provider allows you to send HTTP requests to a specified web service.
- **Creating Client-side actions** - Client-side actions provide a mechanism to launch out from the Cloud Pak for AIOps user interface in the user's browser to any third-party web site.

Creating SSH actions

You can create an action of type script by using the SSH provider. The Operating System of the endpoint system can be either UNIX (including AIX® and Linux®) or Windows. For UNIX, the SSH provider executes a script using the default shell. Alternatively, a shebang (#!) can be used to execute the script in any scripting language the target system can interpret. The SSH provider does not require an additional agent as it uses a direct remote execution connection through SSH.

For Windows, the SSH provider can execute a Powershell script.

Note:

- To access the system through SSH you must set up an SSH server on the Windows system, for example OpenSSH for Windows Server 2019 or Windows 10. The Windows System must have Powershell installed as the default shell of the SSH server.
- If you are using a jumpserver, the jumpserver must be a Linux system. You cannot use a Windows system as a jumpserver.

Complete the following steps to enable script actions:

1. Create an SSH integration. For more information about how to set up an integration, see Configuring {{site.data.keyword.ui_data_integrations}}.
2. Create an action of type SSH. Click **Actions** > **Create action**. Complete the following fields:
    - **Type**:

      Select SSH. If you want to run a script as an action, you must configure an SSH Action Provider on the **Integrations** page.
    - **Name**:

      Provide a name that describes what this action does. For example, the name can be *Find large file systems*.
    - **Prerequisites**:

      If this action requires prerequisites, add this information. For example, `Db2 10.5 or higher`.
    - **Description**:

      Provide any helpful additional information so that the user can immediately understand which problem this action solves.
    - **Tags**:

Provide tags to filter actions. You can select any existing tags from the drop-down list and click the plus icon "+" to add them. To create a new tag, type the text into the field, select the text string displayed, and click the plus icon "+".

- **Shell**:

  Select **Bash** for scripts that run on UNIX systems using `bash` as the ssh shell. Within the script you can specify other script interpreters that are installed on the target system, for example `python`. See the item "Script" below for an example.

  - Notes about how Runbook Automation invokes Bash scripts:
    - It generates a temporary file with the script content.
    - It exports parameters as environment variables.
    - It runs the temporary script file.
    - It deletes the temporary script file.
    - It returns the script return code of the temporary script execution.

  Select **Powershell** for scripts that run on Windows systems using `powershell` as the ssh shell (using `cmd` as the ssh shell is not supported).

  - Notes about how Runbook Automation invokes Powershell scripts:
    - It assigns parameter values to variables with the selected type.
    - It executes the commands from the script.
- **Script**:

  You can either select **Import** to select the script from your file system, or you can directly enter the script into the editor. If you selected **Bash** the script can only be run on UNIX systems using the default shell or the interpreter, which is specified after the shebang `#!`. If you selected **Powershell** the script can only be run on Windows systems using Powershell.

  Notes:

  - Imported scripts must be in UTF-8 format if they contain non-ascii characters (for example, Chinese).
  - If the "Bash" shell type is selected, then any Windows line endings from the imported script are automatically converted to UNIX line endings.

  Examples:

  - Create an SSH action with the following Bash script that uses the `python` interpreter:

    ```
    #!/usr/bin/env python3
    import os
    import platform
    print("Your message \'" + os.environ.get("message") + "\' is presented by
    python version " + platform.python_version())
    ```

    Create a parameter `message` and assign the default value *Hello python* to the parameter. Run the script on a target system where `python3` is installed. The output looks similar to the following output:

    ```
    Your message 'Hello python' is presented by python version 3.9.13
    ```

  - Create an SSH action with the following Bash script that uses the IBM WebSphere Application Server CLI to start, stop, or get the server status of the server:

    ```
    $profilePath/bin/$serverAction.sh $serverName
    ```

    Create the parameters:

    - `profilePath` (the path name of the WAS profile, set the type to a single string, example default value: `/opt/IBM/JazzSM/profile`)
    - `serverAction` (the action to take on the WAS server, set the type to an enumeration of strings (`serverStatus`, `startServer`, `stopServer`), example default value: `serverStatus`)
    - `serverName` (the name of the WAS server, set the type to a single string, example default value: `server1`).

    Run the script on a target system where IBM WebSphere Application Server is installed and has administrative credentials configured in `$profilePath/properties/soap.client.props`.

Alternatively, add parameters for the administrative user name and password and modify the CLI command accordingly.

- **Action output**:
  - **Filter action output** By default, all action output gets collected and printed to the action output field. If you are only interested in a subset of the action output (for example, because you want to create a chain of two actions within a runbook, where some particular information from the first action is used as input to the second action, then you can enable the **Filter action output** toggle and specify a pattern that controls which part of the action output gets returned.
  - **Choose query syntax** Depending on the type of output that gets created you can choose between three types of pattern-matching engines and their respective syntax:
    - **XPath**: best suited for parsing XML data
    - **JSONPath**: best suited for parsing JSON data
    - **Regular expression**: best suited for parsing unstructured data
  - **Choose query syntax at runtime** Select this checkbox if you want to allow the query syntax to be chosen when testing the automation, and also within the parent runbook. This checkbox is typically disabled. If this checkbox is enabled, the system property `filter-syntax` is added to the action.
  - **Enter query pattern** Enter a pattern that is supported by the selected query syntax. For example, if the JSON output object is expected to contain an attribute called "version" and you selected the "JSONPath" query syntax, then you can use the query pattern "$.version" to pick the value of the version attribute out of the overall JSON output.
  - **Enter query pattern at runtime** Select this checkbox if you want to allow the query pattern to be chosen when testing the automation, and also within the parent runbook. If this checkbox is enabled, the system property `filter-query` is added to the action.
  - **Test output filter** Click this link (or the **Test** button in the actions menu) when you are ready to test the action. This saves the action and directly opens the **Test action** page for the current action. For many developers, working with query patterns is an iterative process. Therefore it is recommended to enable **Enter query pattern at runtime** at least while you are developing the action, and switch to **Test action** to fine-tune the query pattern. The development of the query pattern is also supported within the **Test action** page itself, where you can toggle between **Show filtered output** and **Show unfiltered output** after you have run the action. You can also toggle between **Edit action** and **Test action** until you are satisfied with the result.
- **Parameters:**

  Add input parameters to run the action script. Those input parameters are available to the script as environment variables. For example, a parameter `filesystem` is used in a Linux script as `$filesystem`. The `$` sign is automatically added to distinguish user-defined parameters from system parameters such as `target` and `user`. The parameter is exported without the dollar sign. **Note:** input parameter names must consist of alphanumeric characters and the underscore character only. System parameters are not available for use within your own script code. If you want to use some of that data within your script code, you must create related user-defined parameters. The following system parameters exist:

  - **target**: Mandatory parameter. Applies to actions of type Script. The `target` parameter is created to define the target machine where the script is running. By default, it is assumed that the SSH server on the target machine is listening on port 22. If a different SSH port is used, then set the value for the `target` parameter to `hostname:port`, for example, `mytarget.mycompany.com:2022`.
  - **user**: Mandatory parameter. Applies to actions of type SSH. The `user` parameter defines the UNIX or Windows username that is used to run the script on the target machine.
  - **filter-syntax, filter-query**: These system parameters are added if you enable the related checkboxes for action output parsing.

    Note: If the value of the user parameter contains the @ symbol, for example an email address, the @ symbol and all characters that follow it are ignored.

3. Add the action to the runbook, see Adding actions.
4. Run the runbook, see Run your first runbook.

For more information about SSH actions, see the following topics:

- **Single and Multi Target Actions**
- **Output format of Multi Target Actions**
- **Comparison of Single and Multi Target Actions**

Single and multi target actions

You can use single target or multi target actions to execute script actions.

Single Target

The parameter with the name **target** has special meaning to script actions as it defines the system on which the action will run. The system can be defined by a short hostname, a FQDN, or an IP address. When executed a script action will connect to the system identified by the **target** and execute the script there.

The execution result of the script will be reflected in the status of the action. Possible action states are **executing**, **successful**, **unsuccessful**, **failed**, and **unknown**. In the case of a fully automated runbook execution, this status will also be used to decide whether the runbook will be canceled or continued.

Examples:

- An action which receives **prod-server1**, as the content of the target variable will execute this action on the system **prod-server1**. The script exited with return code of zero and the state of the action instance is set to **successful**.
- An action which receives **192.168.55.56**, as the content of the target variable will execute this action on the system with the matching IPv4 adress. The script exited with a non-zero return code and the state of the action instance is set to **unsuccessful**.
- An action which ran in step 1 of a fully automated runbook failed to reach the endpoint (for example because the server was not available) and the action could not be executed at all. The status of the action instance is **failed**. This status is reflected in the status of the execution. The execution will stop after the first step and report a failure.

Multi Target Actions

The other mode of operation of an action is the **Multi Target Action** (MTA). In order to execute an action as an MTA, specify the target string in the following format: **[$target1, $target2, ..., $targetN]**.

The following rules apply to MTAs:

- The target string must begin with a left square bracket (**[**) and must end with a right square bracket (**]**).
- Between the square bracket define a list of systems separated by comma (**,**).
- Duplicated entries are detected and ignored. This means specifying the same target multiple times will have no effect.
- An empty list is also allowed.

Multi Target Actions allow you to execute the same action on any number of targets in parallel, removing the necessity to execute the same action with different targets sequentially. If the target is specified in such a way, how the action is executed and how the results are treated will be different. The following criteria apply to MTAs:

1. For every entry in the comma-separated list, the script will be executed on the specified target. **Note:** you cannot specify the same target more than once.
2. All actions and information are summarized in one log. This log follows a specific format, see Output format of Multi Target Actions.
3. The status of the execution will be **executing** as long as at least one execution is still ongoing and **successful** once all script processes are finished. The result will be **successful** even if some or even all actions reported **unsuccessful**, **failed**, or **unknown**.

4. Specifying an empty array is allowed. No actual execution occurs, but an execution record is still created.
5. The record will always report **successful** as its state. So if the runbook is executed fully automated, the runbook instance will proceed to the next step. This allows the use of multi-target automations as **Conditional Automations**: use the returned lists of previous multi-target automations in subsequent multi-target automations to only run the subsequent automation where the script in the previous automation was successful. Or where it was unsuccessful.

Examples:

- An action receives **[prod-server1]** as the target. It will execute the script on the system **prod-server1**. The script exited with a non-zero return code. The status of the execution is **successful**. The output of the action instance contains information that this script execution was **unsuccesful** on **prod-server1**.
- An action receives **[prod-server1**, **prod-server2**, **prod-server3]** as the target. It will execute the script on three systems in parallel. All scripts exit with zero as the return code. The output and status of each of the three script activities is written to the output. The status of the overall execution is **successful**.
- An action receives **[]** as the target. This triggers the conditional action workflow No execution will take place. The execution record will have an almost empty output and the status will be **successful**.

Output format of multi target actions

Despite being executed on any number of endpoints, the results of multi target actions (MTA) are consolidated into one activity record. This record combines the information from all activities.

The output is formatted as follows:

- If the script produces output the output is displayed after two prefixes. The first item is the name of the target. The second item is either **out** or **err** depending on whether the output was written to `stdout` or `stderr`.
- The log will group the output by target. So the complete output of the first target will be displayed before the output of the second target is displayed and so on. This means the output is not ordered chronologically, although the output by target is.
- At the end of the output a summary is always present. The summary consists of three additional lines. They follow the pattern **$status : [$listOfTargetsWithThisState]**. Where **$status** is one of the following: **unsuccessful**, **successful**, or **failed**.
- If the target array is an empty array, the output will only contain the summary.

Example

An automation has been executed with the target **[prod-server1, prod-server2]**. The execution on **prod-server1** was successful, while the execution on **prod-server2** was unsuccessful. The output will look as follows:

```
prod-server1 out: Some output from the script to stdout happening at 08:34
prod-server1 out: More output from the script to stdout happening at 08:36
prod-server1 out: Script prints more to stdout
prod-server1 status: successful

prod-server2 out: Some output from the script happening at 08:35
prod-server2 out: More output from the script happening at 08:37
prod-server2 err: Script prints output concerning a failure to err
prod-server2 status: unsuccessful

unsuccessful: [prod-server2]
successful: [prod-server1]
failed: []
```

Comparison of single and multi target actions

The following table outlines the difference between the operation modes:

Table. Operation mode differences for target actions

Aspect	Single Target	Multi Target Action
Number of parallel activities	One	Number of entries in target array
Final execution status	Equal to status of automation	Always successful
Failure during fully automated runbook (FARB) execution	Aborts the FARB	Proceeds the FARB
Specifying an empty target	Failure to execute automation	Enters special case for empty array

Creating HTTP actions

The HTTP action provider allows to send HTTP requests to a specified web service. These requests must be sent using the HTTPS protocol.

To create an action of type HTTP, click **Actions** > **Create action** and complete the following fields:

- **Type**

  Select **`HTTP`**.

- **Name**

  Provide a name that describes what this action does. For example, **`IBM Watson Translate`**.

- **Description**

  Provide any helpful additional information so that the user can immediately understand which problem this action solves.

- **Prerequisites**

  If this action requires prerequisites, add this information. For example, **`Watson service credentials are required`**.

- **API endpoint**

  Specify the API endpoint of the web service to which the HTTP request will be sent to. It is mandatory to use the HTTPS protocol, so the API endpoint URL must start with "https://".

  You can choose between a custom API endpoint or one of the supported Cloud Pak for AIOps APIs.

  Note that these supported Cloud Pak APIs automatically handle authentication. Provided that the action is run manually (for example, inside of a manual runbook or via a contextual menu item in the Alert Viewer) by passing through the executing user's information.

  This automatic authentication is not supported in fully automated runbooks. To create an action that calls a supported Cloud Pak API and that can be used in a fully automated runbook, you must explicitly specify an Authorization header as documented for the respective API.

  If you want to access any other Cloud Pak for AIOps API you should do this via the internal hostname **`https://ibm-nginx-svc:443/aiops/api`** instead of using the external hostname. In this case, you must specify an Authorization header as documented for the respective API.

  For example, to retrieve some data from the Metric API, you can use **`https://ibm-nginx-svc:443/aiops/api/app/metric-api/v1/metrics`** with an Authorization header that contains your API (Zen) key. Alternatively, you can use a parameter for the entire URL that contains a full valid web service URL at the time when the HTTP action is started.

- **METHOD**

  Choose the HTTP method. For example **`POST`**.

- **Username**

  If basic authentication is required to use the web service, specify the API user name for basic authentication.

- **Password**

  If basic authentication is required to use the web service, specify the API password for basic authentication.

- **Accept**

  Specify the accept request header. The accept header is used to specify the media types which are acceptable for the response of the request. For example **`text/html`**.

- **Accept-Language**

  Specify the accept-language request header which is used to restrict the set of languages that are preferred for the response of the request. For example **`en-US`**.

- **Additional headers**

  Optionally, specify any additional request headers that are needed for the request. For example **`accept-charset: utf-8`**.

- **Ignore certificate errors**

  Select this check box to ignore certificate errors. Use this option only for test purposes. In production environments, ensure that the correct certificates are installed on the target web service.

- **Action output**

  - **Filter action output** By default, all action output gets collected and printed to the action output field. If you are only interested in a subset of the action output (for example, because you want to create a chain of two actions within a runbook, where some particular information from the first action is used as input to the second action, then you can enable the **Filter action output** toggle and specify a pattern that controls which part of the action output gets returned.
  - **Choose query syntax** Depending on the type of output that gets created you can choose between three types of pattern-matching engines and their respective syntax:
    - **XPath**: best suited for parsing XML data
    - **JSONPath**: best suited for parsing JSON data
    - **Regular expression**: best suited for parsing unstructured data
  - **Choose query syntax at runtime** Select this checkbox if you want to allow the query syntax to be chosen when testing the automation, and also within the parent runbook. This checkbox is typically disabled. If this checkbox is enabled, the system property **`filter-syntax`** is added to the action.
  - **Enter query pattern** Enter a pattern that is supported by the selected query syntax. For example, if the JSON output object is expected to contain an attribute called "version" and you selected the "JSONPath" query syntax, then you can use the query pattern "$.version" to pick the value of the version attribute out of the overall JSON output.
  - **Enter query pattern at runtime** Select this checkbox if you want to allow the query pattern to be chosen when testing the automation, and also within the parent runbook. If this checkbox is enabled, the system property **`filter-query`** is added to the action.
  - **Test output filter** Click this link (or the **Test** button in the actions menu) when you are ready to test the action. This saves the action and directly opens the **Test action** page for the current action. For many developers, working with query patterns is an iterative process. Therefore it is recommended to enable **Enter query pattern at runtime** at least while you are developing the action, and switch to **Test action** to fine-tune the query pattern. The development of the query pattern is also supported within the **Test action** page itself, where you can toggle between **Show filtered output** and **Show unfiltered output** after you have run the action. You can also toggle between **Edit action** and **Test action** until you are satisfied with the result.
- **Parameters**

  Add input parameters to run the action. Those input parameters can be referred in the entry fields. For example, a parameter **`text`** can be referred as **`$text`**.

Creating Client-side actions

Client-side actions provide a mechanism to launch out from the Cloud Pak for AIOps user interface in the user's browser to any third-party website.

The **Client-side** action provider allows you to build a URL from a core URL string and some optional parameters. The URL must use the HTTPS protocol. To run a client-side action, (for example, as an automated step within a runbook) click the **Launch** button. When running a client-side action, the parameters are replaced with the actual values, and the resulting URL is launched in a new browser tab. You can review the URL in the action output area and launch it again by clicking the displayed URL hyperlink.

Client-side actions are composed in the runbook back-end service but they are run in the browser context. Therefore, all runbooks that contain automated steps with client-side actions are considered as manual runbooks. Runbooks that are intended to be run fully automated must not contain client-side actions.

If the third-party website requires authentication, then it will typically ask to provide login credentials before opening the requested page. It is not recommended to use parameters within the client-side action to contain sensitive information (such as user passwords), as they can be seen in clear text in the resulting URL. As a result, password-style string parameters are not supported for client-side actions.

1. To create a client-side action, click **Actions** > **Create action** and complete the following fields:.

- **Type**:

  Select **`Client-side`**.

- **Name**:

  Provide a name that describes what this action does. For example, **`IBM homepage search`**.

- **Description**:

  Provide any helpful additional information so that the user can immediately understand which problem this action solves. For example, **`Search the IBM home page for the given subject`**.

- **Tags**:

  Provide tags to filter actions. You can select any existing tags from the drop-down list and click the plus icon "+" to add them. To create a new tag, type the text into the field, select the text string displayed, and click the plus icon "+".

- **Action client URL**:

  Configure the URL for your action. It is mandatory to use the HTTPS protocol, so the API endpoint URL must start with "https://". You can use parameters. Parameter names must be prefixed with the **`$`** character. For example, **`https://www.ibm.com/search?q=$query`**. Alternatively, specify a parameter that contains a full valid URL of the third-party website at the time when the client-side action is started. For example: **`$openTicketWebPageURL`**.

- **Action parameters**:

  Add input parameters to run the action. Those input parameters can be referred in the URL. For example, a parameter **`query`** can be referred as **`$query`**. The parameter query can be defined as follows:

  - Name: **`query`**
  - Description: **`The subject to be searched on the IBM home page`**
  - Type: **`A single string value`**
  - Minimum length: **`1`**
  - Maximum length: **`not selected`**
  - Password: **`not selected`**
  - Set a default value: **`selected`**
  - Default value: **`runbook`**

2. Add the action to the runbook, see Adding actions.

3. Run the runbook, see Run a runbook.

Ansible actions

The Ansible action provider harvests Ansible type actions from the connected Ansible Automation Platform Controller server. These actions are created automatically and reference an existing asset from a connected Ansible installation.

Ansible Actions can reference the following:

- Job Templates
- Workflow Job Templates

Unlike SSH Actions, Ansible Actions do not define new automation content inside IBM® Runbook Automation. Instead they point to an existing automation, present in the Ansible Automation Platform Controller server, which is a job or job workflow template.

When an Ansible Action runs, it calls the specific "launch" operation of that template on the connected Ansible Automation Platform Controller server. If the template requires prompting for specific variables, these variables are sent by Runbook Automation. Runbook Automation allows these variables to be defined as Action parameters.

Warning: Runbook integration with Ansible Automation Platform 2.5 is not supported by IBM Cloud Pak for AIOps 4.2.0.

See the topics that follow this section for more information about Ansible Actions.

- Harvesting Ansible actions
- Ansible Action interactions
- Job Template parameters and surveys

Job Template parameters and surveys

In Ansible a Job Template or Workflow Job Template can define a number of parameters and properties. They can be defined directly in Ansible's template itself, or can be set to ""prompt on launch"". Every parameter that's set to be prompted must be supplied during template launch by Runbook Automation. In the case of an Ansible action executed through Runbook Automation, this translates to action parameters.

Job Templates in Ansible can accept input for the following parameters:

- diff_mode (type: boolean)
- extra_vars (type: string in JSON format)
- inventory (type: integer)
- credentials (type: integer; enumeration with labels)
- limit (type: string)
- verbosity (type: integer; range 0-5)
- tags (type: string)
- skip_tags (type: string)
- scm_branch (type: string)

Note: Runbook Automation supports only single values for the "credential" parameter. Launching job templates that require more than one credential value to be specified is not supported.

Workflow Job Templates in Ansible can accept input for the following parameters:

- extra_vars (type: string in JSON format)
- inventory (type: integer)

Additionally, you can define any number of additional parameters to be prompted from the user through the "survey". For every parameter defined in the survey an additional Action parameter is created. These parameter names are prefixed with "extraVariables." and have a schema similar to the one defined for the survey parameter. For example, if you define a survey

parameter with the name "age" and define it as an integer with minimum value "0" and maximum value of "99", then the Action will have a parameter called "extraVariables.age" of type "integer", with minimum value of "0" and maximum value of "99".

Note: Float and textarea parameters are not yet supported and will not show up as independent parameters. If you have surveys using these types of survey parameters, enable the "Prompt on launch" flag for the "Extra Variables" selection in your template. That way users can specify the values using the "extraVariables." parameter.

The Ansible Action Provider will harvest the existing Job Templates and Workflow Job Templates and create matching Actions out of them. Those Actions are automatically created and kept synchronized with Ansible. They cannot be changed inside Runbook Automation. To change these Actions, directly edit the Job Template or Workflow Job Template inside Ansible. Runbook Automation will pick up the change and update the Action referencing the asset.

If an Ansible Action changes, make sure to update any parameter mappings made towards that Action in your runbooks.

If a parameter is set to "prompt on launch" in Ansible, but also defines a value, Runbook Automation will import the default value defined in Ansible into the automation parameter.

During execution of Ansible Actions the Runbook Automation UI ensures that you can only pass values that conform to the parameter schemas. If you are using the API, make sure to pass the correct values by comparing your input to the parameter schemas.

Define default values in Ansible for parameters you want to prompt for. For more details on parameters, see the Ansible Documentation for Job Templates or Workflow Job Templates.

Harvesting Ansible actions

Complete the following steps to allow Runbook Automation harvest all Job Templates and Workflow Job Templates and create Ansible actions from those templates.

1. Set up a connection to Ansible.
2. Click **Actions**.
3. Check the Actions for the harvested Ansible Actions.

Ansible action interactions

The Ansible Action Provider automatically converts job templates into actions. These actions are read only in Runbook Automation, but they pick up any change to the original template made in Ansible. To change any aspect of the automation, you can edit the original template.

If you change the Ansible connection, for example to point to another Ansible Automation Platform Controller server with different templates, the previously harvested templates are no longer visible from the Actions page. Runbooks containing those templates can no longer run the actions. Instead, the templates from the new Ansible Automation Platform Controller server are visible.

If you change the Ansible connection back to a previous configuration, the actions that are harvested from this connection reappear, together with their statistics. All runbooks that contain those templates are again able to run those actions.

All runbook instances continue to be fully visible, independent of the state of the Ansible connection.

Predefined actions

Predefined actions perform some standard changes on alerts, like acknowledging the alert, assigning the alert, or modifying the severity of the alert.

Predefined actions are typically invoked from the right-click menu on entries in the Alert Viewer. You can also use predefined actions in the context of your own runbooks, for example to clear the related alert once all steps of the runbook have been performed successfully. Predefined actions may be used in fully-automated runbooks.

If you run an action (either from the actions table or as an automated step within a runbook) and no error occurs, then the resulting success message is not localized. Instead, an English message is displayed, so you can parse the output in a subsequent automated step of your runbook and proceed as appropriate. See the table below for the expected success message for each predefined action.

If you run an action and an error occurs, then whenever possible the output area displays a localized error message, for example: `ARBA1102E Alert with ID '1234' does not exist`. In some cases we receive just a generic message and wrap it, for example: `ARBA1103E Predefined action failed: 400: Bad Request`. In those cases check if the values of the other parameters (like `team` or `owner` for action `Assign owner to alert`) are typed correctly.

Predefined actions are immutable actions that exist after product installation. Each predefined action indicates that the action was last modified by user **System**. The modification date of a predefined action indicates when its source code was created or modified and is independent of the time when the product was installed.

You cannot create additional predefined actions. Similarly, you cannot copy, edit, test, or delete predefined actions.

The following table lists the existing predefined actions.

Predefined actions

Predefined action	Parameters	Expected output
Acknowledge alert	`alert`: ID of the alert to be acknowledged	`Alert acknowledged`
De-Acknowledge alert	`alert`: ID of the alert to be de-acknowledged	`Alert deacknowledged`
Assign owner to alert	`alert`: ID of the alert to be assigned; `team`: Name of the user group to be assigned to the alert (the user who performs this action must be a member of that particular user group; use the value `-` (hyphen) to set the `team` to "unassigned"); `owner`: User ID of the owner to be assigned to the alert (use the value `-` (hyphen) to set the `owner` to "unassigned")	`Alert assigned`
Update severity of alert	`alert`: ID of the alert to be changed; `severity`: New severity (choose from enumeration)	`Alert severity updated`
Suppress alert	`alert`: ID of the alert to be suppressed	`Alert suppressed`
De-Suppress alert	`alert`: ID of the alert to be de-suppressed	`Alert desuppressed`
Clear alert	`alert`: ID of the alert to be cleared	`Alert cleared`

Action versions

You can use versioning to create incremental improvements to actions and monitor the success of changes.

- **Draft**

  The Draft version is a work-in-progress action. It is marked with a draft flag. A draft action is not yet published and does not run in production. The draft is not visible to any operator. You can run and test the current draft.

- **Latest published**

  The latest published version is the version that is running and used by all operators. This is the version that will be used in runbooks where the action is referenced.

- **Archived**

Archived versions are previously used versions. They are not in production and cannot be used by any operations analyst. You can still run archived actions in "Test action".

Save and publish an action

Creating an action involves defining routines that run a task automatically.

After an action is published and used in production, you might improve it based on its success rate. The following actions are available:

- **Cancel:** closes the editor without saving any changes.
- **Save draft:** saves the updates that are added. The editor remains open. **Save draft** is useful if you are editing an action and you want to take a break and continue later. Or you want to save your changes as you go.
- **Save draft & close:** saves your changes and closes the editor. You use this option if you need to work on a different item.
- **Publish:** publishes a new version of the action and makes it available for use in runbooks and for operators.

Testing an action

As a Service Delivery Engineer, you can test an action before you use it within a runbook. This allows you to quickly develop and test actions.

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Actions**.
4. Select an action in the table to display the **Preview** side panel for that action.
5. Click **Test** to start the test procedure for the action. Alternatively, click the options menu of the action and click the **Test** option.

   Note: You can select a different version of the automation if you don't want to test the latest version.
6. Follow the instructions on the **Test action** page.
7. Verify the result of the action.
8. You can test the action multiple times.
9. Parameter values can be modified between test runs of the action. To copy test parameter values to the action editor, click **Edit action**. In the dialog that's displayed, select the test parameter values that you want to bring to the action editor and click **Edit action**. Any changes made in the editor must still be saved.
10. When you are finished testing the action, click **Close**.

Running an action

As an Operator or as a Service Delivery Engineer, you can run the latest published action directly, without wrapping the action into a runbook first. If the action has never been published before, Service Delivery Engineers can also run the draft (although they would typically use **Test action** instead, as the test page is more tightly integrated with **Edit action** and generally allows you to test the latest draft).

1. Click the navigation icon at the upper-left corner of the screen to go to the main navigation menu.
2. In the main navigation menu, click **Operate** > **Automations**.
3. Click **Actions**.
4. Select an action in the table to display the **Preview** side panel for that action.
5. Click **Run action** at the bottom of the side panel to start the run procedure for the action. Alternatively, click the options menu of the action and click the **Run** option.
6. Follow the instructions on the **Run action** page.
7. Verify the result of the action.
8. You can run the action multiple times. Parameter values can be modified between runs of the action.
9. When you are finished running the action, click **Complete**.
10. In the **Provide feedback** section, you can provide a rating and a comment. This information is intended to be evaluated by administrators, so they can easily identify those actions that need to be improved, and understand what exactly needs to be improved.
11. When you are finished with providing feedback, click **Close**.

Activities

In the **Activities** tab on the main page, you can find all actions that have been run, including actions in progress.

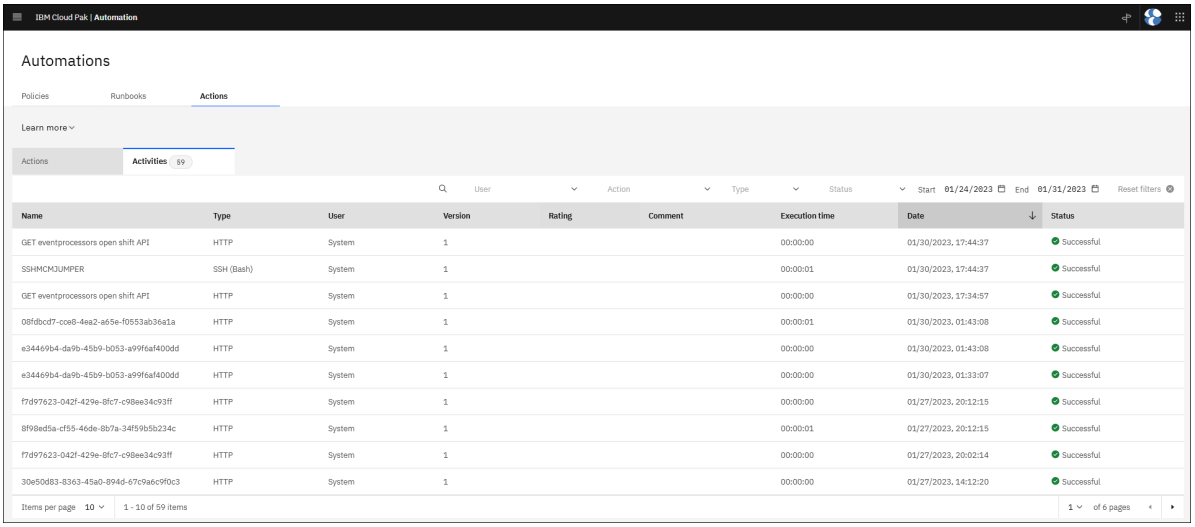


Figure. Activities UI

You can filter your list by the following criteria:

- **User**

  User that ran the action.

- **Action name**

  The action that was run.

- **Type**

  Ansible, HTTP, SSH (Bash), SSH (PowerShell), Client-side, Predefined

- **Status**

Current status of the action activity. Valid statuses are "Executing", "Failed", "Successful", and "Unsuccessful".

The columns **Rating** and **Comment** display the detailed feedback information that has been provided by the operators who have run the action. Service Delivery Engineers can evaluate this feedback to identify those actions that should be improved (for example, as a combination of a high number of invocations and a low average rating, as displayed in the **Actions** table, and the detailed feedback comments on the **Activities** page). Ideally, the feedback comments provided by the operators describe the pain points in sufficient detail, so the Service Delivery Engineer can improve the action efficiently.

Administering IBM Cloud Pak for AIOps

After you install IBM Cloud Pak® for AIOps, you can access several options for administering and customizing the IBM Cloud Pak and IBM Cloud Pak® foundational services.

- CLI tools
- Using a custom certificate
- Secret rotation
- UI console
- Managing user access control
- Managing certificates
- Restarting the Cloud Pak for AIOps cluster
- Backup and restore IBM Cloud Pak for AIOps
- ChatOps insight management
- Infrastructure automation
- Administering data storage
- Updating usage data collection preferences
- Setting the idle session timeout
- Configuring the resources for topology for a production deployment
- Alert and Event schemas
- Configuring probable cause

See the following topics for information about customizing IBM Cloud Pak foundational services:

- Accessing the foundational services endpoint certificate
- Replacing foundational services endpoint certificates
- Refreshing foundational services internal certificates
- Configuring foundational services IAM for single sign-on

CLI tools

Optionally install the command line interface (CLI) tools **`cloudctl`** (deprecated) and **`kubectl`**.

- Installing the IBM Cloud Pak CLI (cloudctl) (deprecated)
- Installing the Kubernetes CLI (kubectl)

Install the IBM Cloud Pak® CLI (cloudctl)

You can use IBM Cloud Pak® CLI (cloudctl) to view information about your cluster, manage your cluster, and more.

After you install IBM Cloud Pak® CLI (cloudctl), you can install another supported CLI, such as the Kubernetes command line tool (kubectl).

To set up the IBM Cloud Pak® CLI, complete the following steps:

1. Get the service hostnames. Use the **management-ingress** service hostname in the command to download the installation file.

   ```
   oc get route -n ibm-common-services
   ```

2. Download the installation file with the following command. Replace the **<cluster_address>** with your IBM Cloud Pak console route.

   ```
   curl -kLo cloudctl-linux-amd64-v3.2.3-1557 https://<cluster address>/api/cli/cloudctl-linux-amd64
   ```

   If you need to get the IBM Cloud Pak console route, run the following command:

   ```
   oc get route -n ibm-common-services cp-console -o jsonpath='{.spec.host}'
   ```

3. Change the file to an executable file and move the file with the following command. Replace **<path_to_installer>** with the path to the directory where you downloaded the cloudctl file, and replace **<install_file>** with the downloaded file name.

   ```
   chmod 755 <path_to_installer>/<install_file>
   ```

   ```
   sudo mv <path_to_installer>/<install_file> /usr/local/bin/cloudctl
   ```

   In this example, it is moved to the **/usr/local/bin/cloudctl** directory.

4. Confirm that cloudctl is installed:

   ```
   cloudctl --help
   ```

   The cloudctl usage is displayed.

5. Log in to your cluster with the following command. Replace **<cluster_address>** with your IBM Cloud Pak console route.

   ```
   cloudctl login -a https://<cluster address> --skip-ssl-validation
   ```

   You can now use cloudctl to view information about your cluster and manage your clusters.

Installing the Kubernetes CLI (kubectl)

You can install and configure the Kubernetes command line tool (kubectl) to complete cluster administrator tasks. Some tasks require the use of both the kubectl tool and the IBM Cloud Pak® CLI (cloudctl).

Before you begin

- Install the IBM Cloud Pak® CLI (cloudctl)
- Synchronize the clocks between your client computer and the nodes in your cluster. To synchronize your clocks, use network time protocol (NTP). For more information about setting up NTP, see the user documentation for your operating system.

Procedure

To set up the kubectl CLI, complete the following steps. You can also download from Kubernetes. For more information, see Install and Set Up kubectl.

1. Get the service hostnames. Use the **management-ingress** service hostname in the command to download the installation file.

   ```
   oc get route -n kube-system
   ```

2. Download the installation file with the following command. Replace the **<cluster_address>** with your IBM Cloud Pak console route.

```
curl -kLo kubectl-linux-amd64-v1.13.11 https://<cluster address>/api/cli/kubectl-linux-amd64
```

If you need to get the IBM Cloud Pak console route, run the following command:

```
oc get route -n ibm-common-services cp-console -o jsonpath='{.spec.host}'
```

3. Change the file to an executable file and move the file with the following command. Replace **<path_to_installer>** with the path to the directory where you downloaded the kubectl file, and replace **<install_file>** with the downloaded file name.

   ```
   chmod 755 <path_to_installer>/<install_file>
   ```

   ```
   sudo mv <path_to_installer>/<install_file> /usr/local/bin/kubectl
   ```

   In this example, it is moved to the **/usr/local/bin/kubectl** directory.

4. Confirm that kubectl is installed:

   ```
   kubectl version --client
   ```

   The kubectl information is displayed.

5. Obtain the cluster configuration details. You can obtain the cluster configuration details by using cloudctl. For more information about the cloudctl commands that you can run, see Managing your cluster with IBM Cloud Pak CLI (cloudctl).

Using a custom certificate

If you want to use your own custom certificate for IBM Cloud Pak® for AIOps or Infrastructure Automation instead of the Red Hat® OpenShift® Container Platform cluster certificate, then complete the following steps after you have installed IBM Cloud Pak® for AIOps or Infrastructure Automation.

1. Export an environment variable containing your project's name.

   ```
   export PROJECT=<project>
   ```

   Where **<project>** is the project that IBM Cloud Pak for AIOps or Infrastructure Automation is deployed in.

2. Ensure that you have created the following four PEM-encoded certificate files.

   - **cachain.pem**: A certificate chain containing the root and intermediate certificates.
   - **carootcert.pem**: The root certificate of your Certificate Authority (CA), with no intermediate or application certificates.
   - **aiops.key.pem**: A key file for the signed certificate in **aiops.pem**.
   - **aiops.pem**: A IBM Cloud Pak for AIOps or Infrastructure Automation certificate, which includes the two fully qualified domain names (FQDNs) for **cpd-${PROJECT}** and **cp-console** in the **Subject Alternative Name** (SAN) list.

   You can find the FQDNs for **cp-console** and **cpd-${PROJECT}** with the following commands.

   ```
   oc get route cp-console -n ibm-common-services -o jsonpath="{.spec.host}"
   oc get route cpd -n ${PROJECT} -o jsonpath="{.spec.host}"
   ```

3. Add your root CA to the CA bundle, to ensure that it is trusted across the Red Hat OpenShift cluster.

   For more information, see the instructions in Replacing the CA Bundle certificate.

4. Create a secret called **customer-tls-secret** with your custom certificates.

   ```
   oc project ${PROJECT}
   oc create secret generic customer-tls-secret --from-file=ca.crt=./cachain.pem --from-file=tls.crt=./aiops.pem --from-file=tls.key=./aiops.key.pem
   ```

5. Restart the **nginx** pods to pickup the new custom certificates.

```
oc delete pod -l component=ibm-nginx
```

6. Patch **ZenService** with the new **customer-tls-secret**.

```
oc patch ZenService/iaf-zen-cpdservice --type merge -p '{"spec":{"zenCustomRoute":
{"route_reencrypt":true,"route_secret": "customer-tls-secret"}}}'
```

7. Patch IBM Cloud Pak® foundational services with the new custom certificate.

```
oc -n ibm-common-services patch managementingress default --type merge --patch '{"spec":
{"ignoreRouteCert":true}}'
oc -n ibm-common-services delete certificates.v1alpha1.certmanager.k8s.io route-cert --
ignore-not-found=true
oc -n ibm-common-services delete secret route-tls-secret
oc -n ibm-common-services create secret generic route-tls-secret --from-
file=ca.crt=./cachain.pem --from-file=tls.crt=./aiops.pem --from-
file=tls.key=./aiops.key.pem
oc -n ibm-common-services delete secret ibmcloud-cluster-ca-cert
```

8. Restart the **auth-idp** and the operand-deployment-lifecycle-manager (ODLM) pods.

```
oc -n ibm-common-services delete pod -l app=auth-idp;oc -n ibm-common-services delete pod
-l name=operand-deployment-lifecycle-manager
```

9. Wait for around 15 minutes for these updates to be reconciled, and for the new configuration to be in place.

   You can check the progress of the reconciliation with the following commands:

```
oc get zenservice iaf-zen-cpdservice -o jsonpath='{.status.Progress}'
oc get zenservice iaf-zen-cpdservice -o jsonpath='{.status.ProgressMessage}'
```

   When **Progress** is shown as *100%*, then reconciliation is complete and you can log in to the console.

Secret rotation

Find out about the secure operation of the following operators, and the procedure you must follow if you update secrets that are owned or used by these operators.

- Issue Resolution Core operator secret usage
- Issue Resolution Lifecycle operator secret usage

Issue Resolution Core operator secret usage

Find out about the secure operation of the **Issue Resolution Core** operator.

Secrets owned by the Issue Resolution Core operator

The **Issue Resolution Core** operator's secrets are created automatically when IBM Cloud Pak® for AIOps is installed. Alternatively, you can manually create some or all of these secrets before IBM Cloud Pak for AIOps is installed, with values of your own choosing. If these secrets exist before IBM Cloud Pak for AIOps is installed, then they are not modified (although new properties might be added).

The following table describes the secrets that the **Issue Resolution Core** operator owns. These secrets contain credentials, encryption keys, and other secrets. Some values can be changed after installation, which might necessitate restarting pods that have access to these secrets, or require extra steps to be run. Details are provided in the table.

Table 1. Secrets that belong to the Issue Resolution Core operator

Secret	Property	Type	Purpose	Rotation	Example commands

Secret	Property	Type	Purpose	Rotation	Example commands
`aiops-ir-core-cem-users-secret`	`clientid`	hex15	OAuth client ID	Change the secret. - Restart `aiops-akora-ui`. - `aiops-ir-core-cem-users`, `aiops-ir-core-esarchiving`, `aiops-ir-core-ncodl-api`, `aiops-ir-core-rba-rbs`, and `aiops-ir-core-usercfg` restart automatically.	`oc set data secret aiops-ir-core-cem-users-secret --from-literal=clientid=$(openssl rand -hex 15) && oc delete pod -l app.kubernetes.io/name=aiops-aiops-akora-ui`
`aiops-ir-core-cem-users-secret`	`clientsecret`	hex15	OAuth client secret	Change the secret. - Restart `aiops-akora-ui`. - `aiops-ir-core-cem-users`, `aiops-ir-core-esarchiving`, `aiops-ir-core-ncodl-api`, `aiops-ir-core-rba-rbs`, and `aiops-ir-core-usercfg` restart automatically.	`oc set data secret aiops-ir-core-cem-users-secret --from-literal=clientsecret=$(openssl rand -hex 15) && oc delete pod -l app.kubernetes.io/name=aiops-aiops-akora-ui`
`aiops-ir-core-cem-users-secret`	`oidcclientid`	hex15	Not used in AIOps	This secret can be changed without effect.	
`aiops-ir-core-cem-users-secret`	`oidcclientsecret`	hex15	Not used in AIOps	This secret can be changed without effect.	

Secret	Property	Type	Purpose	Rotation	Example commands
`aiops-ir-core-cem-users-secret`	`password`	hex15	Basic auth for `https://aiops-ir-core-cem-users:6002`	Change the secret. - Restart `aiops-akora-ui`. - `aiops-ir-core-cem-users`, `aiops-ir-core-esarchiving`, `aiops-ir-core-ncodl-api`, `aiops-ir-core-rba-rbs`, and `aiops-ir-core-usercfg` restart automatically.	`oc set data secret aiops-ir-core-cem-users-secret --from-literal=password=$(openssl rand -hex 15) && oc delete pod -l app.kubernetes.io/name=aiops-aiops-akora-ui`
`aiops-ir-core-cem-users-secret`	`username`	hex15	Basic auth for `https://aiops-ir-core-cem-users:6002`	Change the secret. - Restart `aiops-akora-ui`. - `aiops-ir-core-cem-users`, `aiops-ir-core-esarchiving`, `aiops-ir-core-ncodl-api`, `aiops-ir-core-rba-rbs`, and `aiops-ir-core-usercfg` restart automatically.	`oc set data secret aiops-ir-core-cem-users-secret --from-literal=username=$(openssl rand -hex 15) && oc delete pod -l app.kubernetes.io/name=aiops-aiops-akora-ui`
`aiops-ir-core-couchdb`	`admin_password`	hex15	Basic auth for `https://aiops-ir-core-couchdb-api`	Change the secret and restart `c-example-couchdbcluster-m`.	`oc set data secret aiops-ir-core-couchdb --from-literal=admin_password=$(openssl rand --hex 15) && oc delete pod -l app.kubernetes.io/component=couchdb`
`aiops-ir-core-couchdb`	`couchdb_auth_secret`	hex15	Internal CouchDB use	Change the secret and restart `c-example-couchdbcluster-m`.	`oc set data secret aiops-ir-core-couchdb --from-literal=couchdb_auth_secret=$(openssl rand --hex 15) && oc delete pod -l app.kubernetes.io/component=couchdb`
`aiops-ir-core-couchdb`	`csrf_secret`	hex15	Internal CouchDB use	Change the secret and restart `c-example-couchdbcluster-m`.	`oc set data secret aiops-ir-core-couchdb --from-literal=csrf_secret=$(openssl rand --hex 15) && oc delete pod -l app.kubernetes.io/component=couchdb`

Secret	Property	Type	Purpose	Rotation	Example commands
`aiops-ir-core-couchdb`	`erlang_cookie`	hex15	Internal CouchDB use	Change the secret and restart `c-example-couchdbcluster-m`.	`oc set data secret aiops-ir-core-couchdb --from-literal=erlang_cookie=$(openssl rand --hex 15) && oc delete pod -l app.kubernetes.io/component=couchdb`
`aiops-ir-core-couchdb`	`ibm_password`	hex15	Internal CouchDB use	Change the secret and restart `c-example-couchdbcluster-m`.	`oc set data secret aiops-ir-core-couchdb --from-literal=ibm_password=$(openssl rand --hex 15) && oc delete pod -l app.kubernetes.io/component=couchdb`
`aiops-ir-core-model-secret`	`hkeyname`	ts	Identifier for `hkeyvalue` value	See Key rotation.	
`aiops-ir-core-model-secret`	`hkeyvalue`	hex16	HMAC key for sensitive data in CouchDB	See Key rotation.	
`aiops-ir-core-model-secret`	`oldhkeys`	hex16	Previous HMAC keys for sensitive data in CouchDB	See Key rotation.	
`aiops-ir-core-model-secret`	`keyname`	ts	Identifier for `keyvalue` value	See Key rotation.	
`aiops-ir-core-model-secret`	`keyvalue`	hex32	Encryption key for sensitive data in CouchDB	See Key rotation.	
`aiops-ir-core-model-secret`	`oldkeys`	hex32	Previous encryption keys for sensitive data in CouchDB	See Key rotation.	
`aiops-ir-core-ncodl-api-secret`	`password`	hex15	Basic auth for `https://aiops-ir-core-ncodl-api:10011`	Change the secret. - Restart `cp4waiops-metricsprocessor`. - `aiops-ir-core-ncodl-api` restarts automatically.	`oc set data secret aiops-ir-core-ncodl-api-secret --from-literal=password=$(openssl rand -hex 15) && oc delete pod -l connector.aiops.ibm.com/name=cp4waiops-metricsprocessor`
`aiops-ir-core-ncodl-api-secret`	`username`	hex15	Basic auth for `https://aiops-ir-core-ncodl-api:10011`	Change the secret. - Restart `cp4waiops-metricsprocessor`. - `aiops-ir-core-ncodl-api` restarts automatically.	`oc set data secret aiops-ir-core-ncodl-api-secret --from-literal=username=$(openssl rand -hex 15) && oc delete pod -l connector.aiops.ibm.com/name=cp4waiops-metricsprocessor`

Secret	Property	Type	Purpose	Rotation	Example commands
aiops-ir-core-ncodl-std-secret	password	hex15	Basic auth for https://aiops-ir-core-ncodl-std:10011	Change the secret. -Restart aimanager-aio-ai-platform-api-server, aimanager-aio-change-risk, aimanager-aio-chatops-orchestrator, aimanager-aio-chatops-slack-integrator, aimanager-aio-chatops-teams-integrator, aimanager-aio-controller, aimanager-aio-cr-api, aimanager-aio-log-anomaly-detector, aimanager-aio-log-anomaly-feedback-learning, aimanager-aio-oob-recommended-actions, aimanager-aio-similar-incidents-service. - aiops-ir-analytics-probablecause, aiops-ir-core-api, aiops-ir-core-ncodl-api, aiops-ir-core-ncodl-if, aiops-ir-core-ncodl-jobmgr, aiops-ir-core-ncodl-std, and aiops-ir-core-rba-as restart automatically.	oc set data secret aiops-ir-core-ncodl-std-secret --from-literal=password=$(openssl rand -hex 15) && oc delete pod -l app.kubernetes.io/component=ai-platform-api-server && oc delete pod -l app.kubernetes.io/component=change-risk && oc delete pod -l app.kubernetes.io/component=chatops-orchestrator && oc delete pod -l app.kubernetes.io/component=chatops-slack-integrator && oc delete pod -l app.kubernetes.io/component=chatops-teams-integrator && oc delete pod -l app.kubernetes.io/component=controller,app.kubernetes.io/name=aio && oc delete pod -l app.kubernetes.io/component=cr-api && oc delete pod -l app.kubernetes.io/component=log-anomaly-detector && oc delete pod -l app.kubernetes.io/component=log-anomaly-feedback-learning && oc delete pod -l app.kubernetes.io/component=oob-recommended-actions && oc delete pod -l app.kubernetes.io/component=similar-incidents-service

Secret	Property	Type	Purpose	Rotation	Example commands
`aiops-ir-core-ncodl-std-secret`	`username`	fixed	Basic auth for `https://aiops-ir-core-ncodl-std:10011`	The username is `system`. Do not change, included for future flexibility.	
`aiops-ir-core-omni-secret`	`OMNIBUS_PROBE_PASSWORD`	hex15	Probe password for ObjectServer	See ObjectServer passwords.	
`aiops-ir-core-omni-secret`	`OMNIBUS_ROOT_PASSWORD`	hex15	Root password for ObjectServer	See ObjectServer passwords.	
`aiops-ir-core-postgresql`	`caCertificate`	x509	Root CA certificate for PostgreSQL	This value must be copied from `<instance>-edb-secret`.	
`aiops-ir-core-postgresql`	`dbname`	fixed	Name of event database in PostgreSQL	The database name is `aiops_irb`. This secret must not be changed and is included for future flexibility.	
`aiops-ir-core-postgresql`	`host`	hostname	Service name of PostgreSQL cluster	This value must be copied from `<instance>-edb-secret`.	
`aiops-ir-core-postgresql`	`password`	hex15	Password for PostgreSQL event runtime user	Change the secret. - `aiops-ir-core-ncodl-jobmgr` and `aiops-ir-core-ncodl-std` restart automatically.	`oc set data secret aiops-ir-core-postgresql --from-literal=password=$(openssl rand -hex 15)`
`aiops-ir-core-postgresql`	`port`	port	Port for PostgreSQL cluster service	This value must be copied from `<instance>-edb-secret`.	
`aiops-ir-core-postgresql`	`schema`	fixed	Name of event schema in PostgreSQL	The schema name is `aiops_irb`. This secret must not be changed and is included for future flexibility.	
`aiops-ir-core-postgresql`	`username`	fixed	Name of PostgreSQL event runtime user	The username is `aiops_irb`. This secret must not be changed and is included for future flexibility.	

Secret	Property	Type	Purpose	Rotation	Example commands
`aiops-ir-core-postgresql-admin`	`caCertificate`	x509	Root CA certificate for PostgreSQL	This value must be copied from `<instance>-edb-secret`.	
`aiops-ir-core-postgresql-admin`	`dbname`	fixed	Name of event database in PostgreSQL	The database name is `aiops_irb`. This secret must not be changed and is included for future flexibility.	
`aiops-ir-core-postgresql-admin`	`host`	hostname	Service name of PostgreSQL cluster	This value must be copied from `<instance>-edb-secret`.	
`aiops-ir-core-postgresql-admin`	`password`	hex15	Password for PostgreSQL event admin user	Change the secret. No pod restarts are required.	`oc set data secret aiops-ir-core-postgresql-admin --from-literal=password=$(openssl rand -hex 15)`
`aiops-ir-core-postgresql-admin`	`port`	port	Port for PostgreSQL cluster service	This value must be copied from `<instance>-edb-secret`.	
`aiops-ir-core-postgresql-admin`	`schema`	fixed	Name of event schema in PostgreSQL	The schema name is `aiops_irb`, and must not be changed. It is included for future flexibility.	
`aiops-ir-core-postgresql-admin`	`username`	fixed	Name of PostgreSQL event admin user	The username is `aiops_irb_admin`, and must not be changed. It is included for future flexibility.	
`aiops-ir-core-postgresql-uc`	`caCertificate`	x509	Root CA certificate for PostgreSQL	This value must be copied from `<instance>-edb-secret`.	
`aiops-ir-core-postgresql-uc`	`dbname`	fixed	Name of event database in PostgreSQL	The database name is `aiops_irbuc`, and must not be changed. It is included for future flexibility.	
`aiops-ir-core-postgresql-uc`	`host`	hostname	Service name of PostgreSQL cluster	This value must be copied from `<instance>-edb-secret`.	
`aiops-ir-core-postgresql-uc`	`password`	hex15	Password for PostgreSQL event runtime user	Change the secret. - `aiops-ir-core-usercfg` restarts automatically.	`oc set data secret aiops-ir-core-postgresql-uc --from-literal=password=$(openssl rand -hex 15)`

Secret	Property	Type	Purpose	Rotation	Example commands
aiops-ir-core-postgresql-uc	**port**	port	Port for PostgreSQL cluster service	This value must be copied from **<instance>-edb-secret**.	
aiops-ir-core-postgresql-uc	**schema**	fixed	Name of event schema in PostgreSQL	The schema name is **aiops_irbuc**, and must not be changed. It is included for future flexibility.	
aiops-ir-core-postgresql-uc	**username**	fixed	Name of PostgreSQL event runtime user	The username is **aiops_irbuc**, and must not be changed. It is included for future flexibility.	
aiops-ir-core-postgresql-uc-admin	**caCertificate**	x509	Root CA certificate for PostgreSQL	This value must be copied from **<instance>-edb-secret**.	
aiops-ir-core-postgresql-uc-admin	**dbname**	fixed	Name of event database in PostgreSQL	The database name is **aiops_irbuc**, and must not be changed. It is included for future flexibility.	
aiops-ir-core-postgresql-uc-admin	**host**	hostname	Service name of PostgreSQL cluster	This value must be copied from **<instance>-edb-secret**.	
aiops-ir-core-postgresql-uc-admin	**password**	hex15	Password for PostgreSQL event admin user	Change the secret. No pod restarts are required.	**oc set data secret aiops-ir-core-postgresql-uc-admin --from-literal=password=$(openssl rand -hex 15)**
aiops-ir-core-postgresql-uc-admin	**port**	port	Port for PostgreSQL cluster service	This value must be copied from **<instance>-edb-secret**.	
aiops-ir-core-postgresql-uc-admin	**schema**	fixed	Name of event schema in PostgreSQL	The schema name is **aiops_irbuc**, and must not be changed. It is included for future flexibility.	
aiops-ir-core-postgresql-uc-admin	**username**	fixed	Name of PostgreSQL event admin user	The username is **aiops_irbuc_admin**, and must not be changed. It is included for future flexibility.	

Secret	Property	Type	Purpose	Rotation	Example commands
`aiops-ir-core-rba-devops-secret`	`password`	hex15	Basic auth for `https://aiops-ir-core-rba-as:3080` and `https://aiops-ir-core-rba-rbs:3005`	Change the secret. - Restart `cp4waiops-metricsprocessor`. - `aiops-ir-core-rba-as` and `aiops-ir-core-rba-rbs` restart automatically.	`oc set data secret aiops-ir-core-rba-devops-secret --from-literal=password=$(openssl rand -hex 15) && oc delete pod -l connector.aiops.ibm.com/name=cp4waiops-metricsprocessor`
`aiops-ir-core-rba-devops-secret`	`username`	hex15	Basic auth for `https://aiops-ir-core-rba-as:3080` and `https://aiops-ir-core-rba-rbs:3005`	Change the secret. - Restart `cp4waiops-metricsprocessor`. - `aiops-ir-core-rba-as` and `aiops-ir-core-rba-rbs` restart automatically.	`oc set data secret aiops-ir-core-rba-devops-secret --from-literal=username=$(openssl rand -hex 15) && oc delete pod -l connector.aiops.ibm.com/name=cp4waiops-metricsprocessor`
`aiops-ir-core-rba-jwt-secret`	`secret`	hex15	Key used to sign runbook JSON Web Tokens	Change the secret. - Restart `aiops-akora-ui`, `aiops-ir-core-rba-as`, `aiops-ir-core-rba-rbs`, `connector-orchestrator`.	`oc set data secret aiops-ir-core-rba-jwt-secret --from-literal=secret=$(openssl rand -hex 15) && oc delete pod -l app.kubernetes.io/component=aiops-aiops-akora-ui && oc delete pod -l app.kubernetes.io/component=rba-as && oc delete pod -l app.kubernetes.io/component=rba-rbs && oc delete pod -l app=connector-orchestrator`

Where `<instance>` is the name of your IBM Cloud Pak for AIOps instance, for example `ibm-cp-aiops`.

The `Type` for each secret in the table has one of the following entries:

- hex15, hex16, hex32 - 15, 16, or 32-byte number encoded in hexadecimal (30, 32 or 64 hex digits).
- ts - timestamp value that is used as an identifier for another value. The timestamp is not significant and alternative strings can be used. Some restrictions on the characters might exist. ASCII alphanumeric characters are recommended.
- fixed - do not change this value. The value is included in a secret for future flexibility.
- x509 - an X.509 certificate
- hostname - the DNS hostname of a service.
- port - a TCP port number

Secrets that are used by the Issue Resolution Core operator

The following table describes secrets that are owned by another operator but that are used by the Issue Resolution Core operator. The table shows the Issue Resolution Core components that need to be restarted if these secrets are rotated.

Table 2. Secrets that are used by the Issue Resolution Core operator

Secret	Owner	Purpose	Automatically restarted
`aiops-topology-cassandra-auth-secret`	`asm-operator`	Cassandra contact details	`aiops-ir-core-archiving`, `aiops-ir-core-esarchiving`

Secret	Owner	Purpose	Automatically restarted
<instance>-couchdb-secret	**ibm-aiops-orchestrator**	CouchDB contact details	**aiops-ir-core-cem-users**, **aiops-ir-core-rba-as**, **aiops-ir-core-rba-rbs**
<instance>-es-aiops-user-connection-secret	**ibm-aiops-orchestrator**	Elasticsearch contact details	**aiops-ir-core-datarouting**, **aiops-ir-core-ncodl-api**
<instance>-kafka-secret	**ibm-aiops-orchestrator**	Kafka contact details	**aiops-ir-core-api**, **aiops-ir-core-archiving**, **aiops-ir-core-datarouting**, **aiops-ir-core-esarchiving**, **aiops-ir-core-ncodl-if**, **aiops-ir-core-ncodl-jobmgr**, **aiops-ir-core-ncodl-std**, **aiops-ir-core-rba-rbs**
<instance>-redis-secret	**ibm-aiops-orchestrator**	Redis contact details	**aiops-ir-core-cem-users**, **aiops-ir-core-esarchiving**
internal-nginx-svc-tls	**ibm-zen-operator**	Trusted CA for Zen	**aiops-ir-core-cem-users**
internal-tls	**ibm-zen-operator**	Trusted CA for Zen	**aiops-ir-core-cem-users**
zen-service-broker-secret	**ibm-zen-operator**	Token for accessing Zen API	**aiops-ir-core-cem-users**

Where **<instance>** is the name of your IBM Cloud Pak for AIOps instance, for example **ibm-cp-aiops**.

Key rotation

Warning: The secret **aiops-ir-core-model-secret** contains an encryption key and a HMAC key that is used to protect sensitive data that is stored in CouchDB. Care must be taken when rotating these keys since data that is protected with previous keys cannot be recovered if the keys are lost.

Replacement values for **keyvalue** must be 16 bytes and in hexadecimal format (32 digits). Replacement values for **hkeyvalue** must be 32 bytes and in hexadecimal format (64 digits).

Each key has an associated identifier. Encrypted data is stored along with the identifier of the key that was used to protect it. The identifier must be changed when and only when the key is changed. When the operator creates a new key, it uses the current time as the identifier (for example **2022-06-13T10:58:10Z**). This is a convention but other string formats may be used. If another format is used, there might be some restrictions on the characters that can be used. ASCII alphanumeric characters are recommended.

When rotating keys, all old (key, identifier) pairs must be stored in the **oldhkeys** and **oldkeys** properties so that data that is protected with the old keys can be used. The current key is always used when data is stored or modified. The following examples show the use of the old keys' properties.

1. Secret created by the operator.

```
{
  "hkeyname": "2022-06-13T10:58:10Z",
  "hkeyvalue": "2905c99e41fd6594de9e8632382da185eec94cea545cbd96082552274a454d2b",
  "keyname": "2022-06-13T10:58:10Z",
  "keyvalue": "5cc068f45904c0ff998213941684790e"
}
```

2. Secret after the first rotation.

   The identifiers **myfirsthkey** and **myfirstkey** were used. Time based could have been used.

```
{
  "hkeyname": "myfirsthkey",
  "hkeyvalue": "7539eb64abdae09a500659ca0593140babd00046b175c3a83e3ed28f8d79daca",
  "oldhkeys": "{\"keys\":[{\"name\":\"2022-06-
13T10:58:10Z\",\"value\":\"2905c99e41fd6594de9e8632382da185eec94cea545cbd96082552274a454d
2b\"}]}",
  "keyname": "myfirstkey",
  "keyvalue": "211812c9750778c0175bca4a11cc1e3b",
```

```
  "oldkeys": "{\"keys\":[{\"name\":\"2022-06-
13T10:58:10Z\",\"value\":\"5cc068f45904c0ff998213941684790e\"}]}"
}
```

3. Secret after the second rotation.

   The identifiers **mysecondhkey** and **mysecondkey** were used. Time based could have been used.

```
{
  "hkeyname": "mysecondhkey",
  "hkeyvalue": "07a290a68d6ff7eb45b7febbfd8c5079da5ee40c068f2e1bb87293c22c39b712",
  "oldhkeys": "{\"keys\":[{\"name\":\"2022-06-
13T10:58:10Z\",\"value\":\"2905c99e41fd6594de9e8632382da185eec94cea545cbd96082552274a454d
2b\"},
{\"name\":\"myfirsthkey\",\"value\":\"7539eb64abdae09a500659ca0593140babd00046b175c3a83e3
ed28f8d79daca\"}]}",
  "keyname": "mysecondkey",
  "keyvalue": "f01b36adba59c50841a1444cee8eb09e",
  "oldkeys": "{\"keys\":[{\"name\":\"2022-06-
13T10:58:10Z\",\"value\":\"5cc068f45904c0ff998213941684790e\"},
{\"name\":\"myfirstkey\",\"value\":\"211812c9750778c0175bca4a11cc1e3b\"}]}"
}
```

   If the **oldhkeys** and **oldkeys** properties exist, they must hold the JSON representation of an object with a single property (the name does not matter, this example used **keys**) that has an array of objects. Each of the inner objects has a **name** and a **value** property holding the identifier and value of a previous key.

```
oc delete pod -l app.kubernetes.io/component=rba-as && oc delete pod -l
app.kubernetes.io/component=rba-rbs
```

ObjectServer passwords

Use the following steps to change the ObjectServer password. Ensure that you retain a copy of the original password until these steps are complete.

1. Run the following command to obtain the current passwords from the **aiops-ir-core-omni-secret** secret.

   For example:

```
oc get secret aiops-ir-core-omni-secret -o json | jq -r
'.data.OMNIBUS_PROBE_PASSWORD|@base64d'`
oc get secret aiops-ir-core-omni-secret -o json | jq -r
'.data.OMNIBUS_ROOT_PASSWORD|@base64d'`
```

2. Open a session on the primary ObjectServer pod.

```
oc rsh aiops-ir-core-ncoprimary-0
```

3. Log in to the OMNIbus query tool **nco_sql**, and supply the root password when prompted.

```
$NCHOME/omnibus/bin/nco_sql -u root -s AGG_P_C
```

4. If you want to change the root password, enter the following command.

```
alter user 'root' set password '<NEWPASSWORD>';
go
```

   Where **<NEWPASSWORD>** is the password that you want to set.

5. If you also want to change the probe password, enter the following command.

```
alter user 'probe' set password '<PROBE_PASSWORD>';
go
```

   Where **<PROBE_PASSWORD>** is the password that you want to set.

6. Test the new root password by entering **quit** to leave the SQL prompt, and then log in again and supply the new root password when prompted.

```
$NCHOME/omnibus/bin/nco_sql -u root -s AGG_P_C
```

7. Exit the **oc rsh** session.
8. Wait for 60 seconds to ensure that the password updates have been transferred to the backup ObjectServer.
9. Remote shell into the backup ObjectServer, and test the new root password.

   ```
   oc rsh -c objserv aiops-ir-core-ncobackup-0`
   ```

10. Log in to the OMNIbus query tool **nco_sql**, and supply the root password when prompted.

    ```
    $NCHOME/omnibus/bin/nco_sql -u root -s AGG_B_C
    ```

11. Exit the **oc rsh** session.
12. Update the **aiops-ir-core-omni-secret** with one or more new passwords.

    For example,

    ```
    oc set data secret aiops-ir-core-omni-secret --from-literal=OMNIBUS_PROBE_PASSWORD=
    <PROBE_PASSWORD>
    oc set data secret aiops-ir-core-omni-secret --from-literal=OMNIBUS_ROOT_PASSWORD=
    <NEWPASSWORD>
    ```

    Where

    - **<NEWPASSWORD>** is the new root password
    - **<PROBE_PASSWORD>** is the new probe password
13. If you changed the root password, run the following command to restart **aiops-ir-core-ncodl-if**

    ```
    oc delete pod -l app.kubernetes.io/component=ncodl-if
    ```

Issue Resolution Lifecycle operator secret usage

Find out about the secure operation of the **Issue Resolution Lifecycle** operator.

Secrets owned by the Issue Resolution Lifecycle operator

The **Issue Resolution Lifecycle** operator's secrets are created automatically when IBM Cloud Pak® for AIOps is installed. Alternatively, you can manually create some or all of these secrets before IBM Cloud Pak for AIOps is installed, with values of your own choosing. If these secrets exist before IBM Cloud Pak for AIOps is installed, then they are not modified (although new properties might be added).

The following table describes the secrets that the **Issue Resolution Lifecycle** operator owns. These secrets contain credentials, encryption keys, and other secrets. Some values can be changed after installation, which might necessitate restarting pods that have access to these secrets, or require extra steps to be run. Details are provided in the table.

Table 1. Secrets that belong to the Issue Resolution Lifecycle operator

Secret	Property	Type	Purpose	Rotation	Example commands
aiops-ir-lifecycle-eventprocessor-ep-admin-user	**password**	hex15	Basic authentication for **https://aiops-ir-lifecycle-eventprocessor-ep-jobmanager.aiops.svc:8081**	Change the secret. - **aiops-ir-lifecycle-eventprocessor-ep-jobmanager** restarts automatically.	**oc set data secret aiops-ir-lifecycle-eventprocessor-ep-admin-user --from-literal=password=$(openssl rand -hex 15)**

Secret	Property	Type	Purpose	Rotation	Example commands
`aiops-ir-lifecycle-eventprocessor-ep-admin-user`	`username`	fixed	Basic authentication for `https://aiops-ir-lifecycle-eventprocessor-ep-jobmanager.aiops.svc:8081`	The username is `eventprocessing-admin`. This secret must not be changed and is included for future flexibility.	
`aiops-ir-lifecycle-eventprocessor-ep-internal-keystore`	`password`	hex15	Password for keystore within pods.	Change the secret. - `aiops-ir-lifecycle-eventprocessor-ep-jobmanager` and `aiops-ir-lifecycle-eventprocessor-ep-taskmanager` restart automatically.	`oc set data secret aiops-ir-lifecycle-eventprocessor-ep-internal-keystore --from-literal=password=$(openssl rand -hex 15)`
`aiops-ir-lifecycle-eventprocessor-ep-internal-truststore`	`password`	hex15	Password for truststore within pods.	Change the secret. - `aiops-ir-lifecycle-eventprocessor-ep-jobmanager` and `aiops-ir-lifecycle-eventprocessor-ep-taskmanager` restart automatically.	`oc set data secret aiops-ir-lifecycle-eventprocessor-ep-internal-truststore --from-literal=password=$(openssl rand -hex 15)`
`aiops-ir-lifecycle-eventprocessor-ep-truststore`	`password`	hex15	Password for truststore within pods.	Change the secret. - `aiops-ir-lifecycle-eventprocessor-ep-jobmanager` and `aiops-ir-lifecycle-eventprocessor-ep-taskmanager` restart automatically.	`oc set data secret aiops-ir-lifecycle-eventprocessor-ep-truststore --from-literal=password=$(openssl rand -hex 15)`
`aiops-ir-lifecycle-policy-registry-svc`	`password`	hex15	Basic authentication for `https://aiops-ir-lifecycle-policy-registry-svc:5601`	Change the secret. - Restart `cp4waiops-metricsprocessor`. - `aiops-ir-lifecycle-policy-registry-svc` restarts automatically.	`oc set data secret aiops-ir-lifecycle-policy-registry-svc --from-literal=password=$(openssl rand -hex 15) && oc delete pod -l connector.aiops.ibm.com/name=cp4waiops-metricsprocessor`

Secret	Property	Type	Purpose	Rotation	Example commands
aiops-ir-lifecycle-policy-registry-svc	username	fixed	Basic authentication for https://aiops-ir-lifecycle-policy-registry-svc:5601	The username is system and must not be changed. It is included for future flexibility.	

The Type for each secret in the table has one of the following entries:

- hex15 - 15-byte number encoded in hexadecimal (30 hex digits).
- fixed - do not change this value. The value is included in a secret for future flexibility.

Secrets that are used by the Issue Resolution Lifecycle operator

The following table describes secrets that are owned by another operator but that are used by the Issue Resolution Lifecycle operator. The table shows the Issue Resolution Lifecycle components that are automatically restarted if these secrets are rotated, no manual component restarts are required.

Table 2. Secrets that are used by the Issue Resolution Lifecycle operator

Secret	Owner	Purpose	Automatically restarted
aimanager-ibm-minio-access-secret	AIManager/aimanager	Credentials for Minio S3 storage	aiops-ir-lifecycle-eventprocessor-ep-jobmanager, aiops-ir-lifecycle-eventprocessor-ep-taskmanager
aiops-ir-lifecycle-manager	KafkaUser/aiops-ir-lifecycle-manager	Kafka credentials	aiops-ir-lifecycle-eventprocessor-ep-jobmanager, aiops-ir-lifecycle-eventprocessor-ep-taskmanager
aiops-topology-asm-credentials	ASM/aiops-topology	Credentials for https://aiops-topology-topology.aiops.svc:8080	aiops-ir-lifecycle-eventprocessor-ep-jobmanager, aiops-ir-lifecycle-eventprocessor-ep-taskmanager
aiops-topology-cassandra-auth-secret	ASM/aiops-topology	Credentials for Cassandra	aiops-ir-lifecycle-eventprocessor-ep-taskmanager, aiops-ir-lifecycle-policy-registry-svc
cp4waiops-cartridge-es-auth	Installation	Credentials for Elasticsearch	aiops-ir-lifecycle-eventprocessor-ep-jobmanager, aiops-ir-lifecycle-eventprocessor-ep-taskmanager
iaf-system-elasticsearch-es-ss-cacert-kp	Certificate/iaf-system-elasticsearch-es-ss-ca	CA certificate for Elasticsearch	aiops-ir-lifecycle-eventprocessor-ep-jobmanager, aiops-ir-lifecycle-eventprocessor-ep-taskmanager
<instance>-es-aiops-user-connection-secret	Installation	Binding secret for Elasticsearch	aiops-ir-lifecycle-datarouting, aiops-ir-lifecycle-policy-registry-svc
<instance>-kafka-secret	Installation	Binding secret for Kafka	aiops-ir-lifecycle-datarouting, aiops-ir-lifecycle-eventprocessor-ep-jobmanager, aiops-ir-lifecycle-eventprocessor-ep-taskmanager, aiops-ir-lifecycle-policy-registry-svc
zen-service-broker-secret	ibm-zen-operator	Token for accessing Zen API	aiops-ir-lifecycle-eventprocessor-ep-jobmanager, aiops-ir-lifecycle-eventprocessor-ep-taskmanager

Where <instance> is the name of your IBM Cloud Pak for AIOps instance, for example ibm-cp-aiops.

Administering the UI console

Learn how to access the IBM Cloud Pak Administration panel and IBM Cloud Pak Automation console, and how to use some of the console components.

- Supported browsers
- Accessing the console
- Administration panel
- Customizing the platform

IBM Cloud Pak foundational services Administration panel

The IBM Cloud Pak® foundational services console includes Administration panel that provides an overview of a cluster at a glance to Cluster Administrators. From this UI console, you can view key metrics for specific components of the IBM Cloud Paks and IBM Cloud Pak foundational services that are deployed to the cluster.

As a Cluster Administrator, this at a glance view of key details provides you with the capability to quickly identify and understand anomalies and take related actions to address these issues. Administration panel content is accessible by users with the Cluster Administrator role or the Cloud Pak Administrator role.

Administration panel provides details on all IBM Cloud Paks and IBM Cloud Pak foundational services that are deployed to the cluster and is separate from any overview-related pages that are included with a specific IBM Cloud Pak. For some metrics and components, you can select view more details within a side-panel and go to other UI dashboards, pages, or consoles, such as the Red Hat OpenShift Container Platform web console, to complete associated actions.

Administration panel is available by default and is automatically deployed as a part of installing IBM Cloud Paks and IBM Cloud Pak foundational services. This console can be accessed from opening the main navigation for the console and clicking **Home**. Depending on the IBM Cloud Pak that is installed, you might need to expand an **Administration** category in the main navigation menu and click **Administration Hub** instead of **Home**. The Administration panel can also be accessed from clicking the Cloud Pak switcher (9-dot icon) in the upper-right corner of the toolbar and clicking Administration panel.

Administration panel is composed of three sections:

- The collapsible welcome widget in the console header.
- The quick navigation section below the welcome widget and to the left of the *Overview* section.
- The *Overview* section, which comprises the remainder of the page.

Welcome widget

This collapsible widget displays in the console header to provide you with a list of quick actions that you can take.

The listed quick actions are based on the IBM Cloud Pak foundational services that are deployed to the cluster. This list includes links to the dashboard, page, or side-panel where you can complete associated tasks. This list can include one or more of the following links depending on the services that are installed:

- **Manage identity providers**, which opens the *Identity providers* page of the of the *Identity and Access* dashboard for the Identity and Access Management (IAM) service. You can use this page to configure an LDAP (Lightweight Directory Access Protocol) connection for your product cluster, and to view and edit existing connections.
- **Manage licenses**, which opens the *Licensing* dashboard, where you can view the usage statistics for any deployed IBM Cloud Pak within a selected reporting period. From this dashboard, you can also view, manage, and create reports for licenses that are being used on a cluster.

For more information about the services that are related to any quick action, view the documentation for that service within the documentation for your deployed IBM Cloud Paks or the IBM Cloud Pak foundational services documentation.

Note: Previously this widget included cluster summary and cluster inventory data. This data now displays within separate summary cards in the *Overview* section of Administration panel.

Quick navigation

The *Quick navigation* section includes links for quickly accessing key sites or tools, such as for obtaining support or accessing the IBM Documentation for Administration panel.

Overview

The *Overview* section includes individual summary cards for surfacing the key details for services and components.

The individual summary cards display high-level details for different critical subjects for your deployed IBM Cloud Paks and IBM Cloud Pak foundational services. For your deployed IBM Cloud Pak foundational services, you can view one or more of the following summary cards:

Summary cards

- *Cluster summary*

  The *Cluster summary* card displays the cluster capacity with a breakdown of how much storage memory is used by any deployed IBM Cloud Paks or other deployed objects, and how much memory remains. You can click the card title to open the Red Hat OpenShift Container Platform web console dashboard for the cluster.

- *Cluster inventory*

  The *Cluster inventory* card includes statistics to identify the number of nodes, pods, and persistent volume claims (PVC) on the cluster. Clicking the card title link opens the Red Hat OpenShift Container Platform web console where you can view more details and work with the inventory objects. Clicking each object type opens a different page in the Red Hat OpenShift Container Platform web console:

  - Clicking the **Nodes** link opens the *Nodes* page of the Red Hat OpenShift Container Platform web console.
  - Clicking the **Pods** link opens the *Pods* page of the Red Hat OpenShift Container Platform web console.
  - Clicking the **Persistent Volume Claims** link opens the *Persistent Volume Claims* page of the Red Hat OpenShift Container Platform web console.
- *Cloud Pak deployment*

  Displays the instances of IBM Cloud Paks that are deployed in the cluster. When you click **View details** you can view more details about each instance, including the version of the IBM Cloud Pak that is deployed, the namespace, and the list of pods for the instance. You can click on each pod to open the *Pods* page of the Red Hat OpenShift Container Platform web console to work with the pod.

- *CPU monitoring trends*

  This card shows the current load usage trend for the past five minutes. This card also shows the load average over the past 24 hours and the pod that is responsible for the largest increase in load usage over that period. You can click the pod to open the open the *Pods* page of the Red Hat OpenShift Container Platform web console to work with the pod. You can also click the *View Grafana Dashboard* link to open that dashboard and review additional monitoring data.

- *Workload summary*

  The *Workload summary* card shows a breakdown of the CPU, memory, and network usage by any deployed IBM Cloud Paks. Click **View details** to view a side-panel with additional details. This side-panel displays the three pods with highest utilization of each resource type, **CPU**, **Memory**, or **Network** (network transfer in/out), for the deployed IBM Cloud Paks. You can click any pod to open the OpenShift *Metrics* page of the Red Hat OpenShift Container Platform web console. This page displays related metrics for the resource types.

- *System utility status*

  Displays the status of key services. You can click **View details** or the service status to open the *System utility status* side-panel to view the following details for the installed services:

  - Status for each service
  - Failed pod count (if the services is not running)
  - Restart count (if the services is not running)
  - Failed pods (if the services is not running)

  You can click any listed failed pods to open the page for that pod in the Red Hat OpenShift Container Platform web console. You can also open the main *Pods* page for that web console from this side-panel.

- *Licensing*

  Displays the number of IBM pods that are missing license reporting and the number of products reported. You can click **View details** to open a side-panel that identifies the individual products reported and associated values, such as the number of Virtual Processor Cores (VPC), the Managed Virtual Server (MVS), or other metrics. This panel also lists the specific IBM pods that are missing license reporting.

  Notes:

  - The metric units that are displayed on the side panel represent the current license usage of the product. To view the highest license usage in the reporting period, go to the *Licensing* dashboard.

  If License Service Reporter is deployed on the cluster, you can also click **View license usage report** to open the *Licensing* dashboard. From this dashboard, you can view the usage statistics for any deployed IBM Cloud Pak within a selected reporting period.

- *System events*

  This card shows the latest event notifications for the cluster. You can click **View all** to open the *Events* page of the Red Hat OpenShift Container Platform web console to see all recent notifications.

- *Identity and user access*

  Displays the number of active users and the number of total users. From this card you can click different links to open dashboards to manage user access to the platform and to configure any additional user access that is required to use console features for installed IBM Cloud Paks.

  - The IBM Cloud Pak link within the *Manage platform user access* section of the card opens the *User management* dashboard for adding users, creating and assigning roles and permissions, creating user groups, and more.
  - The **Identity providers** link opens the *Identity providers* dashboard, which you can use to create LDAP connections for user authentication.
  - The **Teams and service IDs** link opens the *Teams* dashboard, which you can use to create and manage your teams. From this view you can also click **Manage service IDs** to open the *Service IDs* dashboard for creating and managing those IDs.

  This summary card was previously known as *User management*.

- *Diagnostics*

  Displays a list of recently generated diagnostics reports, which you can download. You can also select to gather diagnostics and generate a new report. When you click **Gather diagnostics**, a side-panel opens. In this panel, you can select the group or groups of diagnostic statistics that you want to gather into a report. If you select to gather **Cloud Pak data**, you must also specify the namespace where the data is located.

For your deployed IBM Cloud Paks, additional summary cards can be available. For more information about the specific services and subject matter that is shown on these cards, refer to the associated IBM Cloud Pak documentation.

Summary card layout

You can customize the layout of the individual summary cards to make identifying the most critical information for you easier to view quickly.

To rearrange the layout, click the **Move** icon that is in the upper-right of the *Overview* section. You can now drag the individual cards where you want them to display. When you are finished, click **Done**.

Accessing your cluster by using the console

You can access your console from a supported web browser.

Procedure

1. Connect to the URL **`<https://<cluster_address>`**. See the following definitions:

<cluster_address> is the IBM Cloud Pak console route. You can get the IBM Cloud Pak console route by running the following command:

```
oc get route -n ibm-common-services cp-console -o jsonpath='{.spec.host}' && echo
```

Following is a sample output:

```
cp-console.apps.mycluster.mydomain.com
```

Based on the example output, your console URL would be **https://cp-console.apps.mycluster.mydomain.com**.

The username and password are defined in the **config.yaml** file.

2. Access your cluster from a web browser. For a list of supported browsers, see Supported browsers.
3. Log in to the console. You might have different authentication types available for you to select from to authenticate your user credentials.

   The default username to access the console is admin.

   You can get the password for the admin username by running the following command:

   ```
   oc -n ibm-common-services get secret platform-auth-idp-credentials -o jsonpath='{.data.admin_password}' | base64 -d && echo
   ```

   Following is a sample output:

   ```
   EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA
   ```

   Based on the example output, you would use **EwK9dj9fwPZHyHTyu9TyIgh9klZSzVsA** as the password.

   You can change the default password at any time. For more information, see Changing the administrator password.

   Note: Any user with access to the **ibm-common-services** namespace can retrieve this password since it's stored in a secret in the **ibm-common-services** namespace. To minimize password exposure, allow limited users to access the **ibm-common-services** namespace.

Supported browsers

You can access the UI console from Mozilla Firefox, Google Chrome, Microsoft™ Edge, and Safari.

Table. Supported browsers

Platform	Supported browsers
Microsoft Windows	• Microsoft Edge - latest version • Mozilla Firefox - latest version for Windows • Google Chrome - latest version for Windows
Linux®	• Mozilla Firefox - latest version for Linux • Google Chrome - latest version for Linux
macOS	• Mozilla Firefox - latest version for Mac • Google Chrome - latest version for Mac • macOS Safari - latest version

Note: You cannot install the Common Web UI on a system that uses a Windows or Mac operating system. However, you can access the IBM Cloud Pak for AIOps console from a Windows or Mac host by using a browser.

Configuring your logo in the console

You can configure your logo in the following features in the UI console: Login, About modal, and Common header.

Required user type or access level: Cluster administrator or team administrator

Prerequisite

You need the Red Hat® OpenShift® Container Platform CLI (oc) installed. For more information, see Installing the CLI.

Procedure

1. Run the following Red Hat OpenShift Container Platform `oc` command to view and edit your navigation configuration CustomResource:

   ```
   oc edit navconfig common-web-ui-config -n ibm-common-services
   ```

2. Edit the labels section to add a new parameter to override the values:

   Example: `override: "true"`

3. Edit the `header`, `about`, `login` properties. You need to edit `logoUrl`, `logoAltText` for your logo.

   ```
   kind: NavConfiguration
   metadata:
     creationTimestamp: "2019-12-06T17:19:31Z"
     generation: 1
     labels:
       default: "true"
       override: "true"
       name: common-web-ui-config
     name: common-web-ui-config
     namespace: ibm-common-services
   spec:
     header:
       logoUrl: "/common-nav/graphics/mcm-logo.svg"
       logoAltText: "Cloud Pak for Multicloud Management"
     about:
       logoAltText: "Cloud Pak for Multicloud Management"
       copyright: "© 2018 IBM. All rights reserved."
       version: "1.2.3"
       edition: "Enterprise"
     login:
       logoUrl: "/common-nav/graphics/ibm-cloudpak-logo.svg"
       logoAltText: "IBM Cloud Pak"
   ```

Customizing the platform

As an administrator, you can customize the UI console to meet the needs of your enterprise.

Complete the appropriate types of customization for your environment:

- Customizing the branding of the Automation console

  As an administrator, you can customize the Automation console with your company's branding. You can customize the name that is displayed in the Automation console and use a custom logo on your home page.

- Customizing the home page

  As an administrator, you can customize the content that users can see on the home page. You can specify which cards are enabled and which links are displayed in the **Support** section of the home page.

- Configuring your logo in the console

You can use the oc CLI to configure your logo in the following features in the UI console: Login, About modal, and Common header.

Enabling and disabling guided tours

As an administrator, you can customize the IBM Cloud Pak Automation console to enable or disable the *Tours* feature for all users on the platform.

If you do not need the tours to be available to users, you can disable the feature. If you, or another administrator, previously disabled the feature and you want the tours available for your users, you can enable the feature.

Tours are included in the Automation console to help users learn about the IBM Cloud Pak for AIOps platform and features. Some tours introduce high-level concepts, while others offer step-by-step guidance. More tours can be accessed by a user at any time from the Automation console page that is assoicated with the feature to get a better understanding of how to get started with different tasks and components. Users can access available tours by clicking the **Tours** icon in the Automation console toolbar, and clicking their preferred tour.

First-time users are also presented with a guided tour upon login to the Automation console. This tour introduces them to the various home page components, shows how to access tools and capabilities, and where to complete key tasks across the platform.

Before you begin

Required permissions: To customize the home page, you must have one of the following permissions:

- *Administer platform*

Procedure

1. Log in to the IBM Cloud Pak Automation console.
2. From the navigation menu, select **Administration** > **Customizations**.
3. Click **Tours**.
4. Change the setting of the **Enabled**/**Disabled** toggle to either enable or disable the tours feature and to hide the **Tours** icon in the Automation console toolbar.

   Your selection is saved and applied automatically.

Creating custom home page cards with API

Create custom cards to display information on the console home page with the Custom cards REST API.

For example, if you want to show disk usage as a percentage of available disk space, you can add a custom card to the home page that displays the data in a donut or bar chart.

You can use the Custom cards REST API to complete the following tasks:

- Generate an authorization token
- Get a list of custom cards
- Create or replace a custom card
- Edit a custom card
- Delete a custom card

You cannot use the Custom cards API to replace, edit, or delete existing cards that are added by default.

Sample custom cards

You can start with a template to create each of the custom cards that are shown in the following image. Review the template types for a description of the templates that are used.

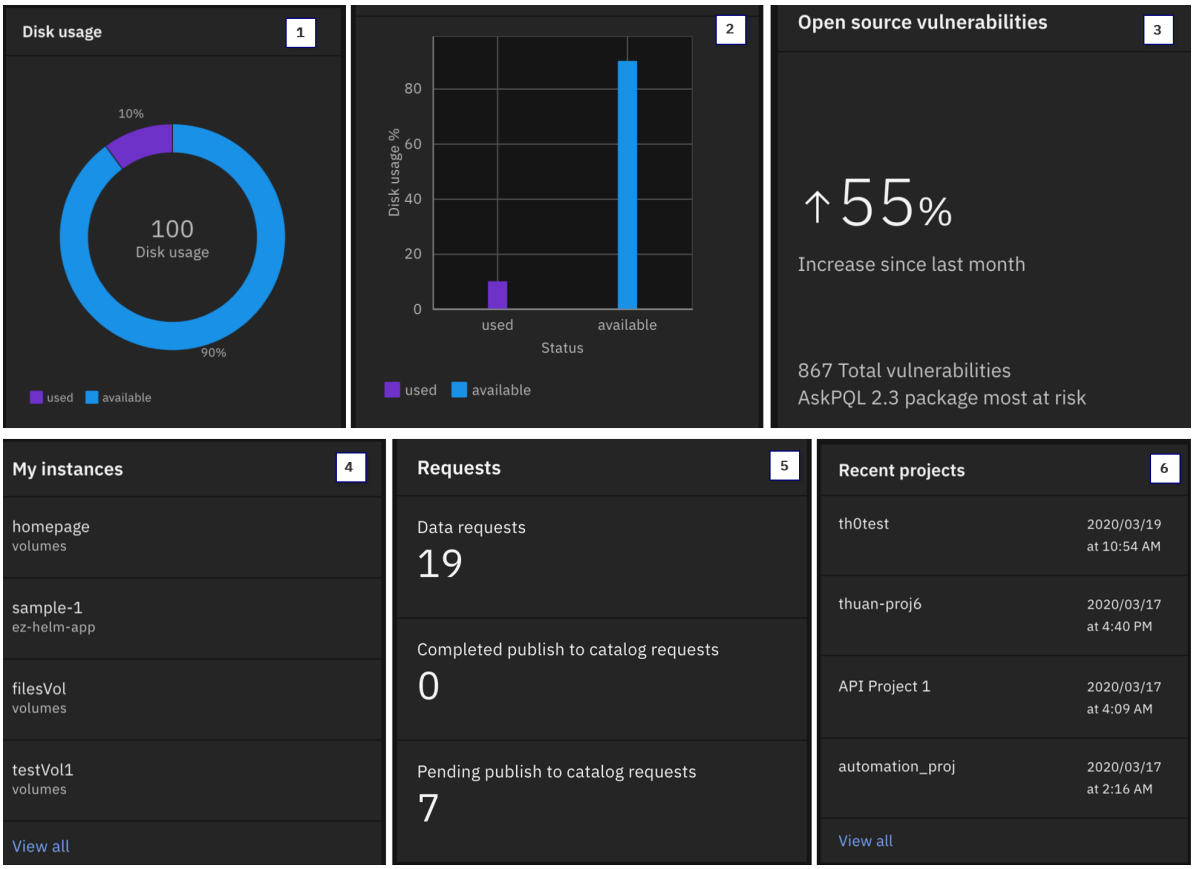


Figure. Custom card types

Template types:

1. The **`donut`** template illustrates an array of values as percentages in a donut chart.
2. The **`bar`** template creates a two-dimensional bar graph.
3. The **`big_number`** template highlights significant data.
4. The **`text_list`** displays a list of text strings.
5. The **`number_list`** displays a list of numeric values and associated text.
6. The **`list`** displays a list of navigation links. The list can contain headers and associated clickable links.

For the JSON payload that describes these custom cards, see the sample command in Create or replace a custom card.

Generate an authorization token

To use the Custom cards REST API, you must first authenticate. Use the token that you get in the response as a token in subsequent API calls.

1. Get an Identity and Access Management (IAM) access token:

```
curl -k -X POST -H "Content-Type: application/x-www-form-urlencoded;charset=UTF-8" \
-d "grant_type=password&username=<username>&password=<password>&scope=openid" \
https://<cluster_address>/idprovider/v1/auth/identitytoken
```

   - **`<username>`** is your username.
   - **`<password>`** is your password.
   - **`<cluster_address>`** is the IBM Cloud Pak console route. You can get the IBM Cloud Pak console route by running the following command:

```
oc get route -n ibm-common-services cp-console -o jsonpath='{.spec.host}'
```

Following is a sample output:

```
cp-console.apps.mycluster.mydomain.com
```

Based on the example output, your console URL would be **https://cp-console.apps.mycluster.mydomain.com**.

The username and password are defined in the **config.yaml** file.

2. Use the IAM access token to call the Platform UI token API:

```
curl -k X GET 'https://<hostname>/v1/preauth/validateAuth' \
-H 'username: admin' \
-H 'iam-token: <iam-token>'
```

- **<hostname>** - The IBM Cloud Pak platform (**cpd**) route hostname.
- **<iam-token>** is the IAM token that you obtained from the response in the previous step.

Get a list of custom cards

Retrieves the properties for the custom cards that you can update.

Curl

```
GET /zen-data/v1/custom_cards
```

Sample command

```
curl -kiv -X GET \
  https://my-deployment-url/zen-data/v1/custom_cards \
  -H 'Authorization: Bearer authorization-token' \
  -H 'Content-Type: application/json'
```

Sample response

In the following code block, the body of the response is replaced with `...`.

```
"_messageCode_": "success",
    "message": "Successfully retrieved custom cards",
    "requestObj": {
      ...
}
```

Create or replace a custom card

Creates a custom card or replaces an existing custom card.

You must include the card definition as a JSON string in the HTTP request body. The definition specifies the structure and the content of the card.

Table. Parameter descriptions

Parameter	Type	Description
key	**String**	The name of the card that you want to create or replace. The string must be unique. If a custom card with the specified key does not exist, the custom card is created, otherwise the existing custom card is replaced.
data	**object**	Provides data to the card. You must provide either a **data_url** object or a **data** object. The **data** object is an object that populates data in the card. The properties of the object must match the type of template defined. The expected structure for each **template_type** parameter is in following examples.

Parameter	Type	Description
`data_url`	`endpoint`	Provides data to the card. You must provide either a `data_url` object or a `data` object. The `data_url` object is an endpoint that is called to populate data in the card. The response that is sent by the endpoint matches the type of template defined. The expected response structure for each `template_type` parameter is shown in the following examples.
`description`	`String`	Helps users understand the purpose of the card. The string must be 200 characters or less.
`drilldown_url` (Optional)	`String`	Defines the URL that the user navigates to when they click **View details**. If this parameter is not defined, the card does not display a **View details** link.
`order`	`integer`	Defines the position of the card. The value must be in the range 1 - 50.
`permissions`	`array`	Specifies the permissions that a user needs to see the card. The array must have at least one permission. If the array has no `permissions` parameter, the card is shown to everyone.
`refresh_rate` (Optional)	`Integer`	The interval in seconds at which the content is refreshed by the source.
`template_type`	`enumerated_type`	Defines the display type of the card. The `template\_type` parameter must be one of the following values: - `donut` - `bar` - `big_number` - `text_list` - `number_list` - `list`
`title`	`String`	The display name for the card. The string must be 40 characters or less.
`window_open_target` (Optional)	`string`	Specifies the target attribute or the name of the window. The following values are supported: - Empty or not provided - the URL gets loaded on the current page by default. - `_blank` - URL is loaded into a new window or tab every time. - `name` - The name of the window where the URL is loaded. The name does not specify the title of the new window.

Use the following examples to understand the `data` object.

bar_data:

```
"data": {
        "bar_data": {
              x_axis_label: 'Packages',
              y_axis_label: 'Projects',
                "data_array": [
                  {
                        "group": "Qty",
                        "value": 65000
                    },
                    {
                        "group": "More",
                        "value": 29123
                    },
                    {
                        "group": "Sold",
                        "value": 35213
                    },
                    {
                        "group": "Restocking",
                        "value": 51213
                    },
                    {
                        "group": "Misc",
                        "value": 16932
                    }
                ]
       }
}
```

big_number_data:

```
"data": {
       "big_number_data": {
              prefix: 'ArrowUp32',
              metric: '1.6',
              suffix: '%',
              sub_text: 'Since last month',
              footer_1: '867 Total vulnerabilities',
              footer_2: 'AskPQL 2.3 package most at risk'
       }
}
```

donut_data:

```
"data": {
              "donut_data": {
                "center_label": "Disk usage",
                "data_array": [
                  {
                    "group": "2V2N 9KYPM version 1",
                    "value": 20000
                  },
                  {
                    "group": "L22I P66EP L22I P66EP L22I P66EP",
                    "value": 65000
                  },
                  {
                    "group": "JQAI 2M4L1",
                    "value": 75000
                  },
                  {
                    "group": "J9DZ F37AP",
                    "value": 1200
                  },
                  {
                    "group": "YEL48 Q6XK YEL48",
                    "value": 10000
                  },
                  {
                    "group": "Misc",
                    "value": 25000
                  }
                ]
              }
            }
```

list_data:

```
"data": {
       list_data: {
              headers: [
                     'project_name',
                     'last_updated_time'
              ],
              rows: [
              {
                     'project_name': 'Improving customer online registration',
                     'last_updated_time': '2 min ago',
                     'nav_url': ' /projects/49753ad5-dcf5-4f94-913f-10acf78a4e71
              },
              {
                     'project_name': '2020 Q1_Increasing Search performance and..',
                     'last_updated_time': '30 min ago',
                     'nav_url': ' /projects/49753ad5-dcf5-4f94-913f-abc
              },
              {
                     'project_name': '2019 Customer shoes sales analysis',
                     'last_updated_time': '1 hour ago',
                     'nav_url': ' /projects/49753ad5-dcf5-4f94-913f-def
              },
```

```
                {
                        'project_name': 'Project X Final version',
                        'last_updated_time': 'Today 11:04 PM',
                        'nav_url': ' /projects/49753ad5-dcf5-4f94-913f-qrs
                },
                {
                        'project_name': 'Project X Final version 2',
                        'last_updated_time': 'Yesterday 12:45 AM',
                        'nav_url': ' /projects/49753ad5-dcf5-4f94-913f-tuv
                }
                ]
        }
}
```

number_list_data:

```
"data": {
        "number_list_data": {
                rows: [
                {
                        drilldown_url: '/zen/#/openSource/packages',
                        label: 'Total approved packages',
                        value: '3'
                },
                {
                        label: 'Total licenses',
                        value: '2345'
                },
                {
                        label: 'Total user interactions',
                        value: '25K'
                }
                ]
        }
}
```

text_list_data:

```
"data": {
        "text_list_data": {
                rows: [
                {
                        drilldown_url: '/zen/#/openSource/packages',
                        label: 'Data virtualization',
                        sub_text: 'dv'
                },
                {
                        label: 'Watson assistant',
                        sub_text: 'assistant'
                },
                {
                        label: 'Watson openscale',
                        sub_text: 'aios'
                }
                ]
        }
}
```

Use the following examples to understand the **data_url** object. The format of the expected response that is provided in the **data_url** from the endpoint for each card type is shown in the following examples.

donut:

```
{
      "center_label": "Disk usage",
      "data_array": [
         {
            "group": "2V2N 9KYPM version 1",
            "value": 20000
```

```
         },
         {
            "group": "L22I P66EP L22I P66EP L22I P66EP",
            "value": 65000
         },
         {
            "group": "JQAI 2M4L1",
            "value": 75000
         },
         {
            "group": "J9DZ F37AP",
            "value": 1200
         },
         {
            "group": "YEL48 Q6XK YEL48",
            "value": 10000
         },
         {
            "group": "Misc",
            "value": 25000
         }
      ]
}
```

bar:

```
{
      "x_axis_label": "Packages",
      "y_axis_label": "Projects",
      "data_array": [
         {
            "group": "Australia",
            "value": 250
         },
         {
            "group": "Brazil",
            "value": 120
         },
         {
           "group": "Canada",
            "value": 190
         },
         {
            "group": "Germany",
            "value": 170
         }
      ]
}
```

big_number:

```
{
       prefix: 'ArrowUp32',
       metric: '1.6',
       suffix: '%',
       sub_text: 'Since last month',
       footer_1: '867 Total vulnerabilities',
       footer_2: 'AskPQL 2.3 package most at risk'
}
```

text_list:

```
{
      rows: [
         {
            label: 'Data virtualization',
            sub_text: 'dv'
         },
         {
            label: 'Watson assistant',
            sub_text: 'assistant'
```

```
         },
         {
            label: 'Watson openscale',
            sub_text: 'aios'
         }
      ]
}
```

number_list:

```
{
      rows: [
         {
            label: 'Total approved packages',
            value: '3'
         },
         {
            label: 'Total licenses',
            value: '2345'
         },
         {
            label: 'Total user interactions',
            value: '25K'
         }
      ]
}
```

list:

```
{
       headers: [
              'project_name',
              'last_updated_time'
       ],
       rows: [
       {
              'project_name': 'Improving customer online registration',
              'last_updated_time': '2 min ago',
              'nav_url': ' /projects/49753ad5-dcf5-4f94-913f-10acf78a4e71
       },
       {
              'project_name': '2020 Q1_Increasing Search performance',
              'last_updated_time': '30 min ago',
              'nav_url': '/zen/#/mydata/datasets'
       },
       {
              'project_name': '2019 Customer shoes sales analysis',
              'last_updated_time': '1 hour ago',
              'nav_url': '/zen/#/mydata/datasets'
       },
       {
              'project_name': 'Project X Final version',
               'last_updated_time': 'Today 11:04 PM',
               'nav_url': '/zen/#/mydata/datasets'
       },
       {
              'project_name': 'Project X Final version 2',
              'last_updated_time': 'Yesterday 12:45 AM',
              'nav_url': '/zen/#/mydata/datasets'
       }
       ]
}
```

Curl

```
PUT /zen-data/v1/custom_cards/{key}
```

Sample command

```
curl -k -X PUT 'https://my-deployment-url/zen-data/v1/custom_cards/unique-card-key' \
--header 'Authorization: Bearer authorization-token' \
--header 'Content-Type: application/json' \
--data-raw '{
  "permissions": ["manage_catalog"],
  "order": 1,
  "title": "Disk usage",
  "template_type": "big_number",
  "data": {      "big_number_data": {
        "metric":  "55",
        "sub_text": "Increase since last month",
        "prefix": "ArrowUp32",
        "suffix": "%",
        "footer_1": "867 Total vulnerabilities",
        "footer_2": "AskPQL 2.3 package most at risk"
      }
    }
}
'
```

Sample response

```
{
    "_messageCode_": "success",
    "data": [
        {
            "id": "537163840404062200",
            "name": "os_vulnerabilities",
            "title": "Open source vulnerabilities",
            "description": "",
            "drilldown_url": "",
            "order": 12,
            "template_type": "big_number",
            "data_url": "",
            "empty_state": {},
            "source_url": "",
            "refresh_rate": 0,
            "data": {
                "big_number_data": {
                    "metric": "55",
                    "sub_text": "Increase since last month",
                    "prefix": "ArrowUp32",
                    "suffix": "%",
                    "footer_1": "867 Total vulnerabilities",
                    "footer_2": "AskPQL 2.3 package most at risk"
                }
            }
        }
    ],
    "message": "PUT custom card succeeded"
}
```

Edit a custom card

Edit the information that is displayed on an existing card.

You can edit the following content:

- `data`
- `data\_url`
- `description`
- `drilldown\_url`
- `order`
- `permissions`
- `template\_type`
- `title`

To edit a custom card, use the rules that are specified in Create or replace a custom card.

Curl

```
PATCH /zen-data/v1/custom_cards/{key}
```

Path parameters

Table. Path parameters

Parameter	Type	Description
key	`String`	The name of the card that you want to update.

Sample command

This command updates the title of a custom card called **`disk_usage_donut`**.

```
curl -kiv -X PATCH \
  https://my-deployment-url/zen-data/v1/custom_cards/disk_usage_donut \

  -H 'Authorization: Bearer authorization-token' \
  -d '{

  "title": "Disk usage"
}'
```

Sample response

```
{
    "_messageCode_": "success",
    "data": [
        {
            "id": "1",
            "name": "homepage_card_disk_usage",
            "title": "Disk usage",
            "description": "",
            "drilldown_url": "/zen/#/v1/diskUsage",
            "order": 2,
            "template_type": "donut",
            "data_url": "/zen-data/v1/homepage/diskUsage",
            "empty_state": {},
            "source_url": "",
            "refresh_rate": 0,
            "data": null
        }
    ],
    "message": "Successfully edited custom card"
}
```

Delete a custom card

Deletes an existing card with the specified key.

Curl

```
DELETE /zen-data/v1/custom_cards/{key}
```

Path parameters

Table. Path parameters

Parameter	Type	Description
key	`String`	The name of the custom card that you want to delete.

Sample command

This command deletes the **`os_vulnerabilities`** card.

```
curl -kiv -X DELETE \
  https://my-deployment-url/zen-data/v1/custom_cards/os_vulnerabilities \
```

```
  -H 'Authorization: Bearer authorization-token'
```

Sample response

```
{
    "_messageCode_": "success",
    "message": "Successfully deleted custom card"
}
```

Customizing the console branding

As an administrator, you can customize the console with your company's branding. You can customize the name that is displayed in the console, and use a custom logo, browser icon, home page and login image.

Before you begin

Required permissions

To customize the branding of the console, you must have one of the following permissions:

- **Administer platform**

About this task

Product name

If you change the product name in the console, the custom name is used throughout the console. For example, the custom name is used on the home page and on the banner of every page in the console.

Logo

If you upload a logo, the logo is displayed in the welcome section of the home page.

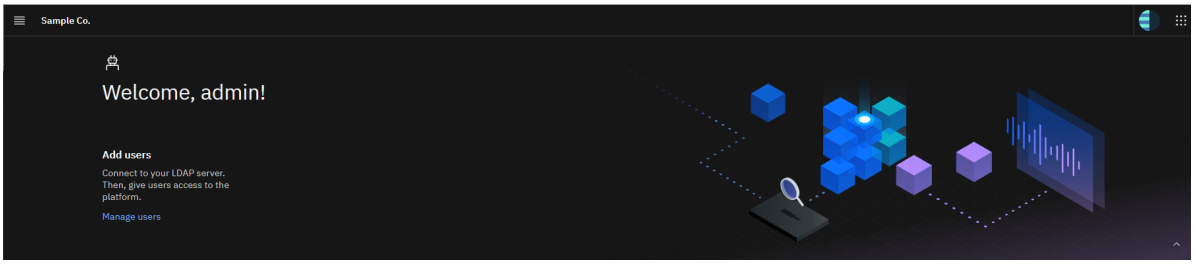


Figure. Console with custom name and logo

Browser icon

If you upload a browser icon, the icon is displayed on the console tab in the browser.

Home page image

If you upload a home page image, the image is displayed on the console home page.

Login image

If you upload a login image, the image is displayed upon login.

Procedure

1. Log in to the console.

2. From the navigation menu, select **Administration** > **Customizations**.
3. Click **Branding**.
4. To use a custom name, select the **Custom name** option under **Product name** and enter the name that you want to use in the console.
    - The name must be 40 characters or less.
    - The product name cannot include the following characters: `<`, `>`, `/`, `\`, `'`, `(`, `)`, or `-`.
5. To use a custom logo, select **Use your own logo** and drop a JPG or PNG file or browse for a file to upload as the home page logo.
    - You can supply a JPG or PNG file.
    - The image is re-sized to fit in a 512 px × 512 px space.
    - The image must be 500 KB or less.
6. To use a custom browser icon, select **Use your own icon** and drop a JPG or PNG file or browse for a file to upload as the browser icon.
    - You can supply a JPG or PNG file.
    - The image is re-sized to fit in a 512 px × 512 px space.
    - The image must be 500 KB or less.
7. To use a custom home page image, select **Use your own image** and drop a JPG or PNG file or browse for a file to upload as the home page image.
    - You can supply a JPG or PNG file.
    - The image must be less than 500 KB for best results, though the limit for an image is 20 MB.
8. To use a custom login image, select **Use your own image** and drop a JPG or PNG file or browse for a file to upload as the login image.
    - You can supply a JPG or PNG file.
    - The image must be less than 500 KB for best results, though the limit for an image is 20 MB.
9. Click **Apply**.

If you want to remove your customizations, click **Restore defaults**.

Customizing the home page

As an administrator, you can customize the content that users can see on the home page. You can specify which cards are enabled and which links are displayed in the **Support** section of the home page.

Before you begin

Required permissions

To customize the home page, you must have one of the following permissions:

- **Administer platform**

About this task

Cards

Cards give users easy access to recently used items and an overview of important changes and alerts.

The list of available cards is determined by the services that are installed on the platform. You can also add custom cards to the home page with the custom cards API. You can prevent certain cards from being displayed on the home page by disabling them.

Each user might see only a subset of the cards based on their permissions and the services that they have access to, so consider how disabling a card will affect users with different roles.

Support links

Support links give users easy access to content that they can use to learn about the platform or to get help.

The default support links are accessible to all users on the platform. However, you can add custom links that are available to specific users based on their permissions.

When you save a change to the home page configuration, the change is immediately applied to the home page. However, if a user is currently logged in to the console, the user must refresh the home page to see the changes.

Procedure

1. Log in to the console.
2. From the navigation menu, select **Administration** > **Customizations**.
3. Click **Home page**.
4. To customize the cards that are displayed on the home page:
    1. Open the **Cards** tab.
    2. Select a card from the list of cards and decide whether you want to display the card on the home page. If you want to suppress the card, toggle the card to **Disabled** and click **Save**.

    Restriction: You cannot add custom cards from the console. You must use the custom cards API.
5. To customize the list of links that are displayed in the **Support** section of the home page:
    1. Open the **Support** tab.
    2. Select a link from the list of links and decide whether you want to display the link on the home page. If you want to suppress the link, toggle the link to **Disabled** and click **Save**.
6. To reorder the list of links that are displayed on in the **Support** section of the home page:
    1. Click the **Reorder** icon .
    2. Drag the links to reorder them.
    3. Click **Save**.
7. To add a link to the list of links that are displayed on in the **Support** section of the home page:
    1. Open the **Support** tab.
    2. Click the **Add custom link** icon .
    3. Specify the label for the link.
    4. Specify the icon to display next to the link.

        You must specify an icon from the Carbon Icon Library.

        By default, when you copy an icon code from the Carbon Design Library, the icon code specifies the 32 px version of the icon. For example, if you copy the code for the Home icon, the following snippet is copied to your clipboard: `<Home32 />`. However, the console uses the 16 px version of the icon. When you paste the snippet into the console:

        1. Remove the angle brackets (`< >`) and the forward slash (`/`).
        2. Change 32 to 16
    5. Specify the URL that you want users to access.

6. Specify who can see this link:
    - If you select **All users**, all platform users can see the link.
    - If you select **Users with any of the selected permissions**, only users with one or more of the selected permissions can see the link.
7. Click **Save**.

Setting your preferences

You can customize the console to work for you.

You can customize your:

- Home page
- Profile photo

Customizing the contents of your home page

You can customize the cards that are displayed on your home page and the links that are displayed in the **Quick navigation** section.

To customize your home page:

1. Click the **Settings** icon:
    - To customize the cards on your home page:
        - Select **Personalize cards**:
            - You can add new cards from the list of **Available cards**.
            - You can drag the cards in the **Overview** section to reorder them.
        - Click **Save and close**.
    - To customize the links in the **Quick navigation**:
        - Select **Personalize navigation**:
            - You can select up to 10 links from the **Add links** list.
            - You can drag the links in the **Quick navigation** section to reorder them.
        - Click **Save and close**.

Changing your profile photo

Your profile photo is displayed next to projects that you create, notifications about actions that you've taken, and more. Your profile photo can make it easier for your coworkers to identify content that you contributed to.

To change your profile photo:

1. Click your avatar in the toolbar.
2. Click **Profile and settings**.
3. Click the **Replace photo** icon  next to the current image.
4. Browse for or drag and drop the image that you want to use. The image can be up to 1 MB and must be a JPG or PNG file.
5. Click **Save**.

You might need to refresh your browser for the change to take effect.

Managing user access control

As an administrator, you are responsible for determining and implementing the best approach for authenticating and managing users.

Identity provider connections

User records are stored in an internal repository database. However, you are recommended to use an enterprise-grade password management solution through identity provider connection, such as an LDAP server for password management. To connect to an identity provider, use the Identity and Access Management (IAM) service. You can open the IAM service from the **Access control** (formerly *User management*) page, by clicking **Identity provider configuration**. For more information about connecting to an LDAP provider with the IAM service, see Identity providers.

If you configure an identity provider connection to an LDAP server, ensure that you grant administrator privileges to a user in your LDAP server.

Important:

- You cannot create a user from the **Access control** page of your cluster Automation console. You can use the **Access control** page to add only an existing LDAP user to your cluster.
- You must use the configured identity provider (LDAP) to manage users that are added from that provider system. For instance, to change or reset the password for a user, you must complete that change in your configured provider system. You cannot change or reset a user password with the Automation console.

  To see your configured identity providers, navigate to **Administration** > **Identity providers** in the Automation console.

Access control

An administrator can manage the permissions that users and groups have on the platform. However, users might need extra permissions.

A user can have multiple roles. The roles can be assigned directly to a user or can be assigned to the user through a user group. If a user has multiple roles, the user has all of the permissions from all of the roles that are assigned to them.

Tip: You can see all of the roles (and permissions) that a user has from the user's profile page, which you can access from the **Access control** page, on the **Users** tab.

If you update a user's role or their group membership and the user is logged in, the user must log out and log back in for the changes to take effect. If the user does not log out, their session will be refreshed after the session times out.

Before you add users to the platform, consider the following questions:

- Do you want to use an LDAP server to manage users' passwords?
- Do you want to use an LDAP server to manage access to the platform?
- Do you want to use user groups to manage users with similar access requirements?
- Do you want to be able to add all of the users in an LDAP group to a user group?
- Do the default roles meet my business requirements?

View the appropriate topic for more information about managing users:

- Roles and permissions
- Adding users
- Managing roles
- Managing user groups
- Setting your preferences
- Generating API keys for authentication
- Configuring LDAP servers and authentication
- Managing namespaces
- Managing Kubernetes Secret passwords

- Changing the cluster administrator credentials

Roles and permissions

For accessing and completing actions with IBM Cloud Pak® for AIOps, users require specific roles and permissions. Review the different available roles, permissions, and the associated actions that users can complete with the roles and permissions. These roles and permissions help you get your users set up to begin day to day operations.

- Predefined roles
- Creating a role
- Permissions

Predefined roles

A role defines the permissions that a user or group has. You can edit the default roles or create new roles if the default set of permissions in a role does not align with your business needs. For more information, see Managing roles.

The following roles are available by default:

- Administrator
- Automation Administrator
- Automation Analyst
- Automation Developer (AIOps Developer role)
- Automation Operator
- Service Administrator
- User

Notes:

- Users with the Service Administrator role do not have permission to add or update a role, or view the details of a user's assign role. If a user with the Service Administrator role selects to view details about a role when viewing details about a user group, a 401 error page is instead displayed.
- The Automation Analyst and User predefined roles are listed as available by default but these roles do not include any permissions for completing actions within IBM Cloud Pak for AIOps. These default roles are used within the IBM® Automation family of offerings, which includes IBM Cloud Pak for AIOps, however, these roles are not used within IBM Cloud Pak for AIOps.

  As these roles do not include or provide any permissions within IBM Cloud Pak for AIOps, do not assign the roles to users within IBM Cloud Pak for AIOps.

  If more IBM Cloud Paks are installed, these roles can have permissions for completing actions within tools for those other IBM Cloud Paks.
- Not all permissions are associated with a predefined role. A user with with a permission to manage roles, such as the Administer platform permission, can assign permissions to a role when needed.

Default user

The default user (`admin`) is automatically assigned the following roles when the roles are added to the platform:

- Administrator

Creating a role

When adding a user, you can choose to create a role instead of assigning an existing role. You can also choose to directly create a role. The options for creating a role are the same regardless of the path you select to create the role.

To directly create a role, complete the following steps:

1. Log in to the console as an administrator with permissions to manage users.
2. From the main navigation, click **Administration** > **Access control**.
3. Select the **Roles** tab and click **New role**.
4. Enter the details for the new role:
    - Name - Enter a distinctive and descriptive name for the role.
    - Description - Enter an optional description for the role to define the role, such as to identify the objective or responsibilities for users with the role.
5. Click **Next**.
6. Select the permissions for the role. You can use the *Find permissions* search field to find and filter the list of permissions.
7. Click **Next**.
8. Review the details for the role and click **Create**.

Permissions

The following table describes the actions that are associated with each permission.

AI models

Table: AI models permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Manage AI models	(1)			X	X	X
View AI models status	(2)		X	X	X	X

1. Manage AI models

   Users with this permission can complete the following actions:

   - Gather and prepare AI training data
   - Train AI models
   - Deploy AI models
2. View AI models status

   Users with this permission can complete the following actions:

   - View AI model status

Resource management (Applications)

Table: Resource management permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Manage applications	(1)		X	X	X	X
Manage resource group templates	(2)			X	X	X
Manage topology comments	(3)	X	X	X	X	X

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
View topologies	(4)	X	X	X	X	X

1. Manage applications

   Users with this permission can manage applications and related settings. Users with this permission can complete the following actions:

   - View applications and resources
   - Create applications
   - Edit applications
   - Delete applications
2. Manage resource group templates

   Users with this permission can manage templates that define how resources are organized into resource groups. Users with this permission can complete the following actions:

   - View resource group templates
   - Create resource group templates
   - Edit resource group templates
   - Delete resource group templates
3. Manage topology comments

   Users with this permission can manage comments on resources included in a topology. Users with this permission can complete the following actions:

   - View topology comments
   - Create topology comments
4. View topologies

   Users with this permission can view application topologies. Users with this permission can complete the following actions:

   - View topologies

Basic operations

Table: Basic operations permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Perform basic operations tasks	(1)	X	X	X	X	X
Use AIOps insights	(2)	X	X	X	X	X

1. Perform basic operations tasks

   Users with this permission can perform basic operations tasks involving applications, resources, alerts, incidents, and runbooks. Users with this permission can complete the following actions:

   - View applications
   - View resources
   - View alerts
   - View incidents
   - Assign incidents
   - Change state of incidents
   - Resolve incidents
   - View published runbooks
   - Run published runbooks
2. Use AIOps insights

Use the AIOps insights dashboard to view productivity metrics such as mean time to restore, incident activity, noise reduction, and runbook usage.

- Use AIOps insights

Data and tool connections

Table: Data and tool connections permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Manage connections	(1)			X	X	X
View connections	(2)		X	X	X	X

1. Manage connections

   Users with this permission can manage connections to infrastructure, log, event, source code, and chat management tools. Users with this permission can complete the following actions:

   - View connections
   - Create connections
   - Edit connections
   - Delete connections
2. View connections

   Users with this permission can view connections to infrastructure, log, event, source code, and chat management tools. Users with this permission can complete the following actions:

   - View connections

Operational policies

This permission is required for viewing policy-related information that can be displayed within the details for an application incident.

Attention: To access **Automations** in the main navigation menu, the *Use runbooks* permission is required in conjunction with the Operational policy roles. For more information, see Runbook permissions.

Table: Operational policies permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Delete operational policies	(1)	X		X	X	X
Edit operational policies	(2)	X		X	X	X
View operational policies	(3)	X	X	X	X	X

1. Delete operational policies

   Users with this permission can delete policies for promoting alerts to incidents. Users with this permission can complete the following actions:

   - Delete operational policies
2. Edit operational policies

   Users with this permission can edit policies for promoting alerts to incidents. Users with this permission can complete the following actions:

- Edit operational policies

3. View operational policies

   Users with this permission can view policies for promoting alerts to incidents. User with this permission can complete the following actions:

   - View operational policies

Platform administration

Permissions in this category enable an administrator to configure, customize, monitor, and manage the platform.

Only the Administer platform permission is associated with a predefined role (Adminstrator). This permission includes all other platform administration permissions.

Table: Platform administration permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Administer platform	(1)					X
Manage configurations	(2)					
Manage platform health	(3)					
View platform health	(4)					

1. Administer platform

   This permission offers the most comprehensive set of actions for managing and monitoring the platform. Users with this permission have elevated privileges and can grant or revoke all permissions, including other administrative permissions. Users with this permission can complete the same actions as users that have the following permissions:

   - Manage configurations (Administer platform)
   - Manage platform health (Administer platform)
   - Manage platform roles (User administration)
   - Manage users (User administration)
   - Manage user groups (User administration)
   - Manage service instances (Service instances)

2. Manage configurations

   Users with this permission can customize the platform, integrate the platform with other applications, and enable connections to unsupported data sources. Users with this permission can access the Customizations page, the Configurations page, and the JDBC drivers tab on the Platform connections page. Some actions require specific services to be installed. Users with this permission can complete the following actions:

   - Configure connection to SMTP server

     Note: Requires Watson Studio or Watson Knowledge Catalog. An SMTP connection enables the platform to send emails.

   - Configure integration with IBM Guardium appliances

     Note: Requires Watson Knowledge Catalog. Use IBM Guardium to audit access to sensitive data on remote databases.

   - Configure connections to Hadoop clusters

     Note: Requires Execution Engine for Apache Hadoop.

   - Customize branding

   - Enable and disable home page cards

- Enable and disable default support links
- Add and delete custom support links
- Enable and disable guided tours
- Import JDBC drivers

  Note: Requires the common core services. JDBC drivers enable users to connect to unsupported data sources.

3. Manage platform health

   Users with this permission can monitor resource use, set quotas and alerts, manage workloads to maintain the health of the platform, and gather diagnostic data when problems occur. Users with this permission can access the Monitoring page and the Diagnostics page. Users with this permission can complete the following actions:

   - Monitor workloads and resource use

     Note: Workloads include active runtimes and jobs. Resources include memory and vCPU use across services, instances, and environments.

   - Stop any runtime environment

     Note: Environments are hardware and software configurations defined for running analytical assets or jobs.

   - View pod status, details, and logs

     Note: Services are composed of Kubernetes pods. A pod is an instance of a process that runs on the cluster.

   - Restart pods
   - View platform quotas and service quotas

     Note: A quota specifies the maximum amount of memory or vCPU that the platform or a service should use.

   - View event history and alerts
   - Set and edit platform resource quotas
   - Set and edit individual service resource quotas
   - Create and run diagnostics jobs

     Note: A diagnostic job gathers information that can be used for troubleshooting problems.

   - Delete diagnostics jobs

4. View platform health

   Users with this permission can monitor resource use and workloads across the platform to gauge the health of the platform. Users with this permission have read-only access to the Monitoring page. Users with this permission can complete the following actions:

   - Monitor workloads and resource use

     Note: Workloads include active runtimes and jobs. Resources include memory and vCPU use across services, instances, and environments.

   - View pod status, details, and logs

     Note: Services are composed of Kubernetes pods. A pod is an instance of a process that runs on the cluster.

   - View platform quotas and service quotas

     Note: A quota specifies the maximum amount of memory or vCPU that the platform or a service should use.

   - View event history and alerts

Runbooks

Table: Runbooks permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Administer runbooks and runbook application	(1)			X	X	X
Author and manage runbooks	(2)			X	X	X
Author runbooks	(3)	X		X	X	X
Use runbooks	(4)	X	X	X	X	X

1. Administer runbooks and runbook application

   Users with this permission can complete the following actions:

   - View draft and published runbooks
   - Run draft and published runbooks
   - Create draft runbooks
   - Edit draft runbooks
   - Approve draft runbooks
   - Delete draft and published runbooks
   - Edit connections
   - Edit global settings
   - Edit group ownership
2. Author and manage runbooks

   Users with this permission can complete the following actions:

   - View draft and published runbooks
   - Run draft and published runbooks
   - Create draft runbooks
   - Edit draft runbooks
   - Approve draft runbooks
   - Delete draft and published runbooks
3. Author runbooks

   Users with this permission can complete the following actions:

   - View draft and published runbooks
   - Run draft and published runbooks
   - Create draft runbooks
   - Edit draft runbooks
4. Use runbooks

   Users with this permission can complete the following actions:

   - View published runbooks
   - Run published runbooks

Secure tunnel

Table: Secure tunnel permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Install tunnel connectors	(1)			X	X	X

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Manage tunnel applications	(2)			X	X	X
Manage tunnel connections	(3)			X	X	X
Manage tunnel templates	(4)			X	X	X
Renew certificates	(5)			X	X	X
View tunnel audit logs	(6)			X	X	X

1. Install tunnel connectors

   Users with this permission can install tunnel connectors on remote clusters or hosts. Users with this permission can complete the following actions:

   - Install tunnel connectors
2. Manage tunnel applications

   Users with this permission can manage application traffic at the address or port level. Users with this permission can complete the following actions:

   - View application mappings
   - Create application mappings
   - Edit application mappings
   - Delete application mappings
3. Manage tunnel connections

   Users with this permission can manage connections to remote clusters or hosts. Users with this permission can complete the following actions:

   - View tunnel connections
   - Create tunnel connections
   - Edit tunnel connections
   - Delete tunnel connections
4. Manage tunnel templates

   Users with this permission can manage templates that simplify the creation of tunnel connections. Users with this permission can complete the following actions:

   - View tunnel templates
   - Create tunnel templates
   - Edit tunnel templates
   - Delete tunnel templates
5. Renew certificates

   Users with this permission can renew certificates for tunnel connections. Users with this permission can complete the following actions:

   - Renew certificates for tunnel connections.
6. View tunnel audit logs

   Users with this permission can view a history of user-driven operations and tunnel connection events. Users with this permission can complete the following actions:

   - View tunnel audit logs

Service instances

A service instance is a specific deployment of a service. Some services can be deployed more than once. Some service instances have their own access controls.

Table: Service instances permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Create service instances	(1)					X
Manage service instances	(2)					

1. Create service instances

   Users with this permission can create service instances and storage volumes. The types of service instances depend on the services that are installed. Users with this permission can complete the following actions:

   - Create service instances

     Note: If an instance has access controls, the user who creates the instance is an instance administrator.

2. Manage service instances

   Users with this permission can manage access to any service instance or delete any service instance from the Instances page. Users with this permission can complete the following actions:

   - Create service instances
   - View all service instances
   - Add users to any service instance
   - Assign an instance role to instance users
   - Remove users from a service instance
   - Delete any service instance

Topology administration

Table: Topology administration permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Manage advanced topology settings	(1)			X	X	X
Manage topology presentation	(2)			X	X	X
Manage topology rules	(3)			X	X	X
Manage topology tools	(4)			X	X	X

1. Manage advanced topology settings

   Users with this permission can define global preferences for creating and using topologies. Users with this permission can complete the following actions:

   - View advanced topology settings
   - Edit advanced topology settings

2. Manage topology presentation

   Users with this permission can define how topologies are presented to end users. Users with this permission can complete the following actions:

   - View topology presentation settings
   - Create topology presentation settings
   - Edit topology presentation settings
   - Delete topology presentation settings

3. Manage topology rules

   Users with this permission can define how resources are processed to enable event correlation, tagging of resources, and the creation of end-to-end topologies. Users with this permission can complete the following actions:

   - View topology rules
   - Create topology rules
   - Edit topology rules
   - Delete topology rules
4. Manage topology tools

   Users with this permission can define custom actions for resources that are included within topologies. Users with this permission can complete the following actions:

   - View topology tools
   - Create topology tools
   - Edit topology tools
   - Delete topology tools

User administration

User administration permissions enable an administrator to manage users, groups, and roles. These permissions apply to the platform. Service instances and workspaces such as projects, catalogs, and deployment spaces have their own access controls.

Table: User administration permissions

Permissions	Actions	Automation Developer	Automation Operator	Service Administrator	Automation Administrator	Administrator
Manage platform roles	(1)					
Manage user groups	(2)			X		
Manage users	(3)			X		

1. Manage platform roles

   Users with the *Manage platform roles* permission can access the Roles tab on the Access control page to modify platform roles or create custom roles. Roles determine the permissions that a user or user group has. This permission does not apply to service instances or assets, such as projects, catalogs, and deployment spaces. Users with this permission can complete the following actions:

   - Create platform roles
   - Edit platform roles
   - Delete platform roles
2. Manage user groups

   Users with the *Manager user groups* permission can access the User groups tab on the Access control page to manage user groups. User groups make it easier to manage the roles (and permissions) of users with similar access requirements. Users with this permission can complete the following actions:

   - Create user groups
   - Edit user groups
   - Delete user groups
   - Assign roles to user groups
   - Remove roles from user groups
3. Manage users

   Users with the *Manager users* permission can access the Users tab on the Access control page, and can onboard users to the platform. Users with this permission can complete the following actions:

   - Add users
   - Edit user profiles
   - Assign roles to users

- Remove roles from users
- Remove users

What permissions do I have?

You can see what permissions you have from your profile. Your permissions are determined by the roles that are assigned to you.

To see what roles are assigned to you:

1. Log in to the IBM Cloud Pak Automation console and click your avatar in the upper right of the toolbar.
2. Click **Profile and settings**.
3. Click the **Roles** tab.

The permissions that are associated with your role (or roles) are listed in the **Enabled permissions** column.

Adding users

To add users, you need to have an external LDAP server set up for creating user profiles and authentication. When a user record is included within the LDAP server, you can add the user account within the IBM Cloud Pak Automation console. You can add a user directly or as part of an LDAP group. For more information, see Adding users.

Assigning roles and permissions

When you add a user or group, you must specify the role and permissions that they have. These roles and permissions can be granted to users through a combination of the Automation console and an external LDAP server for managing user authentication.

Red Hat OpenShift Container Platform roles

To add and manage users and roles for accessing and using the Red Hat OpenShift Container Platform web console, you need to use the Red Hat role-based access control (RBAC). For more information, see the Red Hat OpenShift topic Using RBAC to define and apply permissions.

Adding users

When you add a user to the platform for accessing the IBM Cloud Pak Automation console and tools, a user profile (or record) is created for the user.

You can add a user to the platform in the following ways:

- You can give a user from a connected LDAP service access by directly assigning that user a role.
- You can include a user from a connected LDAP service into a user group. When you add a user to a user group, the platform automatically assigns the user each role that is assigned to the group.
- You can give all members of an LDAP group access to the platform by adding that LDAP group to a user group.

When a user is given access to the platform, a profile is created for the user. This profile lists the user's roles and permissions. The user can also now access the Automation console.

All users that are added to the platform are also automatically included within a default *All users* group. This group is used to give all users access to the Automation console and common features, such as their profile and settings. You cannot edit or delete this group.

Important:

- You cannot create a user from the **Access control** page of your cluster Automation console. You can use the **Access control** page to add only an existing LDAP user to your cluster.
- You must use the configured identity provider (LDAP) to manage users that are added from that provider system. For instance, to change or reset the password for a user, you must complete that change in your configured provider system. You cannot change or reset a user password with the Automation console.

  To see your configured identity providers, navigate to **Administration** > **Identity providers** in the Automation console.

Before you begin

- To manage access to the Automation console, you must have one of the following permissions:
  - **Administer platform**
  - **Manage users**
- To add users to IBM Cloud Pak® for AIOps, you need to connect to an LDAP service and use that connection to create the users.

  Ensure that you set up an LDAP service as an identity provider. To configure an LDAP connection, you must be an Administrator with permissions to manage users. For more information, see Configuring an LDAP connection.

  Any user that needs access to IBM Cloud Pak® for AIOps must exist in your LDAP service with their **`user name`** and **`email`** attributes set. If these attributes are not set for a user, you might not be able to find and add the user to IBM Cloud Pak® for AIOps.
- To add a user, you need at least one of the following types of information about the user from your connected LDAP service:
  - The user's full name. If you use this information when adding a user, you must enter the name exactly as it is specified in the LDAP service.
  - The username for the user. This value maps to the field that you specified for the **User search field** in the LDAP configuration
  - Email address. If you use this information when adding a user, enter the email address exactly as it is specified in the LDAP service.

Procedure

1. Log in to the IBM Cloud Pak Automation console as an administrator with permissions to manage users.
2. From the main navigation, click **Administration** > **Access control**.
3. From the *Users* tab, click **Add user**.
4. Search for the user's profile information by entering one of the following details about the user:
   - User's name (full name)
   - Username
   - Email

   Limitations:
   - You cannot select a specific LDAP service to use for adding users. All connected LDAP services are searched for the specified user ID, username, or email.
   - You cannot use an external lookup service for finding and adding users from the LDAP service to IBM Cloud Pak for AIOps.
   - You need to know the user ID, user name, or email of a user within the connected LDAP service to find and add that user.
5. Click **Next**.
6. Set the platform access permissions for the user. Privileges on the platform are controlled by permissions. Users are granted permissions through role assignment. Users can be assigned roles directly or inherit roles from user groups that they are included within. Users within a user group inherit all roles and permissions that associated with the user group.

Select either of the options to set permissions for the user:

- **Assign roles directly**

  Select this option to directly assign one or more existing roles or create a role for the user. For more information about existing roles and permissions or about creating a role, see Roles and permissions.

  - Optional. Create a new role to assign to the user.
  - Select the checkbox for each role that you want to assign to the user. If you created a new role for the user, you must still select the checkbox for that role.
  - Click **Next**.

- **Add to user group**

  Select this option to add a user to an existing user group or groups. For more information about user groups, including how to create a group, see Managing user groups.

  - Select the checkbox for each user group or groups that you want to include the user within.
  - Click **Next**.

7. Review the *Summary* of the user profile, user groups, roles, and access. When finished, click **Add**

What to do next

- Edit the user details and roles, such as to add more roles or to add the user to a user group. When you are editing a user, you can complete the following changes:
  - View, add, and remove roles
  - View, add, and remove permissions
  - Add the user to a user group
  - Remove the user from a group
  - Change the user full name
  - Change the user email
  - Delete the user from IBM Cloud Pak for AIOps

  You can also add a user group, such as to add and manage roles for all users that you include in the group.

- Login as the newly added user

  As a user directly added from the LDAP service or from an added user group, log in to the console and verify that the expected roles and permissions exist.

  Upon login, either your *Home* page or the *Administrator panel* is displayed. Users with an administrator role view the *Administrator panel* upon login, while all other users view their *Home* page. From the main navigation menu, browse through your available tools and pages to ensure that you can view the correct data and complete the tasks for your assigned permissions.

Managing roles

If the default roles that are provided do not meet your needs, you can create new roles or edit the permissions that are associated with the default roles.

Before you begin

Required permissions

To manage roles, you must have one of the following permissions:

- *Administer platform*

About this task

You can create and edit roles from the **Roles** tab of the **Access control** page.

Procedure

To create a role:

1. Log in to the console.
2. From the navigation menu, select **Administration** > **Access control**.
3. Open the **Roles** tab.
4. Click **New role**.
5. Enter a name and, optionally, a description for the role.
6. Click **Next**.
7. From the **Permissions** section, select the permissions that you want to associate with the role.
8. Click **Next**.
9. Review the summary. If the values are correct, click **Create**.

Results

You can now assign the role to users and groups.

Managing user groups

You can create user groups to simplify the process of managing large groups of users.

User groups make it easier to manage a large number of users with similar access requirements. For example, if you know that 20 different users are going to collaborate on a project and they all need the same role, you can add them to a group that is assigned that role. If a member of the group leaves the company, you can remove the user from the group, rather than looking for all of the assets that the user has access to.

Before you begin

Required permissions

To manage groups, you must have one of the following permissions:

- **Administer platform**
- **Manage groups**

About this task

You can create and edit groups from the **User groups** tab of the **Access control** page.

Important:

- From foundational services version 3.19 (Platform UI version 1.6.0), when creating a user group, you can either create assigned groups or dynamic groups (new option).

If you select the **Assigned** option, you can select the users you want to include.

If you select the **Dynamic** option, you can create attribute-based rules to determine which users are automatically added to the group.

- By default, an **All users** group is included. As the name suggests, all users are automatically included in this group. The group is used to give all platform users access to the console. You cannot edit or delete this group.

Procedure

To create a user group using the Assigned option:

1. Log in to the console.
2. From the navigation menu, select **Administration** > **Access control**.
3. Open the **User groups** tab.
4. Click **New user group**.
5. Enter a name and, optionally, a description for the role.
6. Under Membership type, select **Assigned**.

   Note: With assigned membership, you select the users that you want to add to the group.
7. Click **Next**.
8. Specify the users to include in the group. The available options depend on whether the environment is connected to use an identity provider for authentication, such as an LDAP server.

   - **Existing users** - Find and add existing platform users.
   - **Identity provider users** - Find and add a user from a connected identity provider, such as an LDAP server.
   - **Identity provider groups** - Find and add a user group from a connected identity provider, such as an LDAP server. If an IAM integration is enabled, choose this option.

   After you find the user or group, select the user or group to add them to the new user group. Repeat this action to add more users or groups.

   Notes:

   - When searching for a user, you can search by the user's name, email, or username.
   - If you have the **Manage users** permission and you do not see the user that you want to add to the group, first create the user. For more information, see Managing console access.
   - If you add users from an LDAP group, the users might not immediately be added to the user group. If you do not see a user within the group, log in to the console as that user. Upon login, the Platform UI detects that the user belongs to an LDAP group and creates the user profile. User profiles for LDAP group users are created only when the users log in to the console.
9. Click **Next**.
10. Select one or more roles that you want to assign to this group.
11. Click **Next**.
12. Review the summary. If the values are correct, click **Create**.

To create a user group using the Dynamic option:

1. Log in to the console.
2. From the navigation menu, select **Administration** > **Access control**.
3. Open the **User groups** tab.
4. Click **New user group**.

5. Enter a name and, optionally, a description for the role.
6. Under **Membership type**, select **Dynamic**.

   Note: With dynamic membership, you create attribute-based rules to determine which users are automatically added to the group.
7. Click **Next**.
8. In **Membership rule**, you create attribute-based rules to determine which users are automatically added to the group.

   Notes:

   - You can specify one or more rules.
   - You can either require that all of the rules are true or that at least one of the rules is true for the user to be included.
   - There are four **Attributes** that you can select:
     - Location
     - Nationality
     - Organization
     - User Type
   - There are four **Operators** that you can select:
     - Equal
     - Not equal
     - Match
     - Not match
9. Once you select the attributes and operators, type in the values based on values from your identity provider.
10. Click **Next**.
11. Select at least one role to assign to the new user group. For example, you can assign **Administrator** and/or **User** to the user group. Alternatively, you can **create a new role** and assign the new role you created.
12. Click **Next**.
13. Review the summary. If the values are correct, click **Create**.

Results

You can now use the group to give users access to various assets on the platform.

Configuring LDAP servers and user authentication

The Identity and Access Management (IAM) for platform users includes authentication that includes OIDC and SAML, and authorization that includes user management, and Cloud Resource Naming (CRN).

Lightweight Directory Access Protocol (LDAP) support

IBM Cloud Pak for AIOps can be configured with a single or multiple LDAP servers for the authentication and authorization. IBM Cloud Pak for AIOps supports the following LDAP types:

- IBM Tivoli Directory Server
- IBM Lotus Domino
- IBM SecureWay Directory Server
- Novell eDirectory
- Sun Java™ System Directory Server
- Netscape Directory Server

- Microsoft Active Directory
- Custom

Configuring LDAP connections

With IBM Cloud Pak for AIOps, you can authenticate across multiple LDAPs. You can add multiple directory entries to the LDAP config in the **`server.xml`** file. Liberty automatically resolves the domain name from the login and authenticates against the targeted LDAP directory. IBM Cloud Pak for AIOps users and user groups are associated with an enterprise directory during the time of the user and user group onboarding through import. When the new LDAP directory entry is created, the domain name also gets added as a new entry. At the time of login, you can specify the domain against which the authentication can be validated.

It is possible to have a mix of directory types, for example AD, Tivoli, and OpenLDAP. Role-based access control (RBAC) is enforced on the LDAP domain. Cluster administrators and Cloud Pak administrators have access to all LDAP domains, whereas team administrators are restricted to only those domains they are authorized to.

For more information, see Configuring LDAP connection.

- Multiple LDAP domains
- Authentication types
- Configuring single sign-on
- Configuring LDAP connections
- Changing LDAP cache settings
- Changing LDAP search settings
- Changing LDAP search cache variable values
- Updating SCIM LDAP attributes mapping
- Changing Logjam configuration property

Multiple LDAP domains

There is a growing need for IBM Cloud Pak® for AIOps users to be able to authenticate across multiple LDAPs. Sometimes large organizations might have an LDAP domain controller for different global regions or subsidiaries.

Users can have a mix of directory types such as AD, Tivoli, OpenLDAP etc.

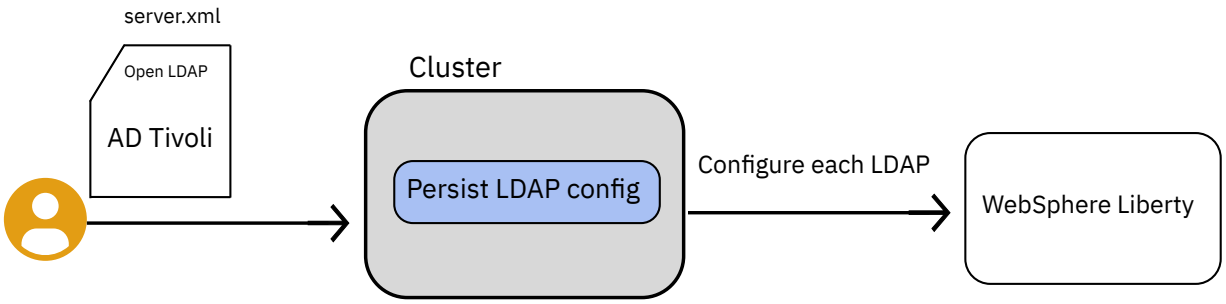
Users can configure multiple directories in the LDAP configuration in IBM Cloud Pak for AIOps. IBM Cloud Pak for AIOps uses WebSphere Liberty Server OpenID Connect as an authentication service which does administration and authentication against the appropriate directory.

Note:

- Currently, IAM doesn't support LDAP failover.
- Currently, the nested LDAP group is not supported in IAM.

Multiple LDAP registration

As a cluster administrator, you can configure multiple LDAP domains by adding multiple directory entries to the LDAP configuration in **`server.xml`**.



In a multiple domain configured environment, a new user administration on IBM Cloud Pak for AIOps platform enforces a selection of appropriate domains and the user is added to the Team.

The user profile and the domain name is maintained by IBM Cloud Pak for AIOps and is further used for user management. The ability to chose domain before selecting users, such as for a team, allows administrator to isolate teams with a specific domain.

Note: User credentials are passed by IBM Cloud Pak for AIOps to WebSphere Liberty OIDC server which resolves the user domain and authenticates the user with a matching domain.

Authentication types

You can configure multiple authentication types for single sign-on (SSO).

You can then access your IBM Cloud Pak® console by using the configured authentication type.

The following authentication types are supported:

- IBM provided credentials (admin only). The default authentication type that you configure during IBM Cloud Pak® installation. This configuration is usually the Administrator username and password. The foundational services installer generates a password for the default username `admin`, which is a cluster administrator role.
- Enterprise LDAP
- Enterprise SAML
- OpenShift authentication
- OpenID Connect

After you install your IBM Cloud Pak®, when you access the console, you can see the login options that are available. You can see the login options only for the authentication types that are configured in your cluster. If you configure only the default authentication type (IBM provided credentials), you do not see any login option. You must then access your IBM Cloud Pak® console by using the default authentication type.

The console login cookie saves the authentication type that you select for 24 hours. When you access the login page within 24 hours, you see the login page for the same authentication type. You can choose another authentication type by clicking **Change your authentication type**.

Note: If you remove the Enterprise LDAP or disable Enterprise SAML configuration from your cluster, first clear the browser cache before you access the console login page.

Setting the preferred login options

When you configure multiple authentication types in your cluster, all the configured types are displayed on the console login page as authentication options.

If you want the console login page to display any one or a selected set of the configured login options, you can set the preferred login options.

To set the preferred login options before IAM service installation, see setting the preferred login options in Configuring foundational services by editing the CommonService custom resource.

To set the preferred login options after IAM service installation, complete these steps:

1. Log in to your infrastructure node by using the oc login command.
2. Edit the **platform-auth-idp** configmap.

   ```
   oc edit cm platform-auth-idp --n ibm-common-services
   ```

   In the data section, you see the **PREFERRED_LOGIN** data definition, which has no values set by default.
3. Add the login options that you want to display on the cosole page. For example, **PREFERRED_LOGIN: "SAML","LDAP","OIDC"**.

   Use the following parameter values:

   - For default authentication, use **DEFAULT**
   - For enterprise LDAP, use **LDAP**
   - For enterprise SAML, use **SAML**
   - For OpenShift authentication, use **ROKS**
   - For OpenID Connect, use **OIDC**
4. Restart the **auth-idp** pod by deleting it.

   1. Get the auth-idp pod name.

      ```
      oc get pods -n ibm-common-services | grep auth-idp
      ```

      Following is a sample output:

      ```
      auth-idp-785df784f5-qcx4z                         4/4     Running   0          39d
      ```

   2. Delete the auth-idp pod.

      ```
      oc delete pod <auth-idp-pod-name> -n ibm-common-services
      ```
5. Restart the **common-web-ui** pod by deleting it.

   1. Get the common-web-ui pod name.

      ```
      oc get pods -n ibm-common-services | grep common-web-ui
      ```

   2. Delete the **common-web-ui** pod.

      ```
      oc delete pod <common-web-ui-pod-name> -n ibm-common-services
      ```

After the pods restart, your preferred login options are visible on the console login page.

Using foundational services IAM for authentication

You can log in to the Red Hat® OpenShift® Container Platform console by using your foundational services or the IBM Cloud Pak access credentials. For more information, see Configuring foundational services IAM for single sign-on.

Delegating authentication to OpenShift

Authenticate with Red Hat® OpenShift® Container Platform.

If you are installing IBM Cloud Pak for AIOps on an environment that uses the built-in OpenShift OAuth server, you can use the OpenShift OAuth server for authentication.

Authentication with OpenShift is enabled by default. You can disable it, if required.

Note: If you are upgrading your cluster, the authentication configuration is not changed after the upgrade. For example, if you did not configure OpenShift authentication in the previous release, the OpenShift authentication is not enabled after you upgrade.

Updating OpenShift authentication before you install the IAM Operator

Complete the following steps to update the OpenShift authentication parameters.

You must complete these steps after you install the IBM Cloud Pak foundational services operator but before you install the foundational services in your cluster. For more information about installing IBM Cloud Pak foundational services, see IBM Cloud Pak foundational services Installer service.

1. From your Red Hat® OpenShift® Container Platform console, click **Operators** > **Installed Operators**.
2. Search for **IBM Cloud Pak foundational services** and click the **IBM Cloud Pak foundational services** operator name to open the Details tab for the operator.
3. Select the **Common Service** tab. You see an instance of the CommonService that is installed in your cluster.
4. Edit the **`common-service`** instance.
5. In the YAML file, scroll down to the **`spec.services.ibm-iam-operator.spec.authentication.config`** section.

   Note: Create the **`spec.services.ibm-iam-operator.spec.authentication.config`** section if it does not exist.

   The following example shows the **`spec.services.ibm-iam-operator.spec.authentication.config`** section in the YAML file:

```
apiVersion: operator.ibm.com/v3
kind: CommonService
metadata:
  name: common-service
  namespace: ibm-common-services
spec:
size: ...
  ...
  services:
   - name: ibm-iam-operator
     spec:
       authentication:
        config:
         roksEnabled: false
         roksURL: <endpoint url>
         roksUserPrefix: 'IAM#'
```

6. Update the values of the following parameters:

   - **`roksEnabled:`** Set to **`false:`** to disable authentication with Red Hat® OpenShift® Container Platform. The default value is **`true`**.
   - **`roksURL:`** The public service endpoint URL of your public cloud cluster. Use the following command to get the endpoint URL.
   - **`roksUserPrefix:`** Prefix to be used with the username. When you access your cluster console or CLI, you use the prefix along with the username to authenticate with OpenShift. The default value is **`""`**. If you are using IAM with Red Hat® OpenShift® on IBM Cloud® service in the IBM Cloud, you must set the prefix to **`IAM#`**.

   Note: You need **`jq`** to run the following command. To install **`jq`**, see Download jq.

   ```
   curl -sk https://<server-URL>/.well-known/oauth-authorization-server | jq -r '.issuer'
   ```

   For example,

   - Code sample: **`curl -sk https://api.par-test.os.fyre.ibm.com:6443/.well-known/oauth-authorization-server | jq -r '.issuer'`**
   - Output of the code sample: **`https://oauth-openshift.apps.par-test.os.fyre.ibm.com`**

   The following shows a configuration of the **`ibm-iam-operator`** with the OpenShift authentication configuration in the **CommonService** YAML file:

   ```
   - name: ibm-iam-operator
       spec:
         authentication:
           config:
             roksEnabled: true
             roksURL: 'https://oauth-openshift.apps.par-test.os.fyre.ibm.com'
             roksUserPrefix: 'IAM#'
         oidcclientwatcher: {}
         pap: {}
         policycontroller: {}
         policydecision: {}
         secretwatcher: {}
         securityonboarding: {}
   ```

7. Save the YAML file.

8. Add the OpenShift user from the IBM Cloud Pak for AIOps console **Administration** > **Access control** page.

When the user accesses the IBM Cloud Pak for AIOps console, they can now see the **OpenShift authentication** as a login option.

Changing the default Cloud Pak Administrator

The IBM Cloud Pak foundational services installation creates a default **`admin`** user, who is a cluster administrator. If you are delegating authentication to OpenShift, you can customize the default OpenShift username by using the **`bootstrapUserId`** parameter. This user will have the Cloud Pak Administrator access.

To add the **`bootstrapUserId`** parameter after IAM service installation, complete these steps:

1. Log in to your infrastructure node by using the **`oc login`** command.

2. Edit the **`platform-auth-idp`** configmap.

   ```
   oc edit cm platform-auth-idp --n ibm-common-services
   ```

   In the **`data`** section, the **`"BOOTSTRAP_USERID"`** is set to **`kubeadmin`** by default. Change **`kubeadmin`** to the OpenShift user who you want to make the Cloud Pak Administrator.

3. Restart the **`auth-idp`** pod by deleting it.

   1. Get the **`auth-idp`** pod name.

      ```
      oc get pods -n ibm-common-services | grep auth-idp
      ```

      Following is a sample output:

      ```
      auth-idp-785df784f5-qcx4z                                 4/4     Running   0          39d
      ```

2. Delete the **auth-idp** pod.

```
oc delete pod <auth-idp-pod-name> -n ibm-common-services
```

After the pod restarts, the OpenShift user can access the cluster with Cloud Pak Administrator privileges.

Updating OpenShift authentication after you install the IAM operator

If you installed the **ibm-iam-operator** in your cluster with OpenShift authentication enabled, and you want to disable OpenShift authentication or you want to update the parameters, complete the following steps:

1. Follow the steps in Updating OpenShift authentication before you install the IAM Operator to either provide the OpenShift authentication parameters or to disable the authentication.
2. Restart the following pods. You can restart the pods by deleting them from your Red Hat® OpenShift® Container Platform console **Workloads** > **Pods** page.
   - **auth-pdp**
   - **common-web-ui**
   - **auth-idp**
   - **oidcclient-watcher**

Clear the browser cache and access IBM Cloud Pak for AIOps console login page.

IAM integration with Red Hat® OpenShift® on IBM Cloud® service in the IBM Cloud

If you are using IAM with Red Hat® OpenShift® on IBM Cloud® service in the IBM Cloud, there are some known limitations to consider:

- The **cloudctl login** command does function for OpenShift IBMid users. These users might need to log in to the console to obtain the Kubectl context for completing CLI operations, such as running **cloudctl** commands.
- Searching for groups in a Red Hat® OpenShift® on IBM Cloud® service environment to add users to a team is not supported. You can search for only users in an Red Hat® OpenShift® on IBM Cloud® service environment to add to a team for authorization purposes. Even after an LDAP group synchronization, you can search for only users from an Red Hat® OpenShift® on IBM Cloud® service environment with IAM for adding users to teams. You cannot search or add groups.
- Delegating authentication to multiple providers that use different **user_prefix** values at the same time is not supported. For a Red Hat® OpenShift® on IBM Cloud® service in the IBM Cloud cluster, you can delegate authentication to either the default IBMid provider or the LDAP provider, but not both as the **ROKS_USER_PREFIX** value is used for all users.
- For user authorization, add users to teams either by searching for the users, or set rolebinding or clusterrolebinding for the users on the Red Hat® OpenShift® on IBM Cloud® service environment in the IBM Cloud platform.
- Configuring the OpenShift server URL as the **ROKS_URL** might be a different procedure across OpenShift versions.

Configuring foundational services IAM for single sign-on

You can use the IBM Cloud Pak foundational services Identity and Access Management (IAM) service for single sign-on (SSO) to access the Red Hat® OpenShift® Container Platform console.

Required user type or access level in foundational services: Cluster administrator **Required user type or access level in Red Hat® OpenShift® Container Platform:** kubeadmin

The Cluster Administrator can configure the IAM service as an OpenShift OAuth identity provider. Cluster users can then log in to the Red Hat® OpenShift® Container Platform console by using their IBM Cloud Pak foundational services or IBM Cloud Pak console access credentials.

Prerequisites

- You must have access to the Red Hat® OpenShift® Container Platform console as `kubeadmin`.
- You must install the foundational services IAM operator in your cluster.

Prepare for configuring foundational services IAM as the identity provider

Gather the information that is required to configure the identity provider.

- Get the default cluster administrator username and password, which the IAM service sets during installation.
  - To get the default username, run the following command:

    ```
    oc -n ibm-common-services get secret platform-auth-idp-credentials -o
    jsonpath='{.data.admin_username}' | base64 --decode
    ```

  - To get the default password, run the following command:

    ```
    oc -n ibm-common-services get secret platform-auth-idp-credentials -o
    jsonpath='{.data.admin_password}' | base64 --decode
    ```

- Get the IBM Cloud Pak foundational services console URL.

  ```
  oc get route -n ibm-common-services cp-console -o jsonpath='{.spec.host}'
  ```

- Get the Red Hat® OpenShift® Container Platform console URL. You use this URL as the `post_logout_redirect_uris` and `trusted_uri_prefixes` parameter values when you Define an OIDC `Client` resource.

  ```
  oc -n openshift-console get route console -o=jsonpath={.spec.host}
  ```

- Get the Red Hat® OpenShift® Container Platform console OAuth URL. You can find the URL when you click the `Copy Login Command`. The `Copy Login Command` is on the Red Hat® OpenShift® Container Platform console in the `kube:admin` drop-down list.

  1. Click **kube:admin** > **Copy Login Command**.
  2. Copy the URL that is displayed in the address bar. You need only the `https:<url-content>.com` portion of the URL. For example, `https://oauth-openshift.apps.eying.com`.

  You use this URL in the `redirect_uris` parameter value when you Define an OIDC `Client` resource.

- Get the issuer certificate authority (CA) root certificate.
  - You can get the certificate from the Red Hat® OpenShift® Container Platform console.
    1. Navigate to **Networking** > **Routes** page.
    2. Click your IBM Cloud Pak foundational services console route. You see the **cp-console** page.
    3. In the **Overview** tab, scroll down to the **CA Certificate** text box.
    4. Copy the content that is in the **CA Certificate** text box.
    5. Paste the content in a file and save it as a .pem on your local machine.

  (OR)

  - You can run the following command to get the certificate:

    ```
    oc -n ibm-common-services get secret icp-management-ingress-tls-secret -
    o=jsonpath='{.data.ca\.crt}' | base64 --decode | tee -a cert.pem
    ```

Register the foundational service IAM OIDC client

Complete these steps to register the IAM service as an OIDC client:

1. Define an OIDC `Client` resource. Create a `ocpclient.yaml` file with the following content:

   ```
   apiVersion: oidc.security.ibm.com/v1
   kind: Client
   metadata:
   ```

```
  name: ocpclient
  namespace: default
spec:
  secret: ocpclientsecret
  oidcLibertyClient:
    post_logout_redirect_uris:
    - '<Red Hat® OpenShift® Container Platform console URL>:443'
    redirect_uris:
    - '<OpenShift console OAuth URL>/oauth2callback/IBM-IAM'
    trusted_uri_prefixes:
    - '<Red Hat® OpenShift® Container Platform console URL>:443'
```

You must update the following parameters that are in the resource:

- **post_logout_redirect_uris** and **trusted_uri_prefixes**: Add your Red Hat® OpenShift® Container Platform console URL and port 443. See console URL.
- **redirect_uris**: Add your Red Hat® OpenShift® Container Platform OAuth URL. See console OAuth URL. You must append **/oauth2callback/IBM-IAM** to the URL. **IBM-IAM** is the endpoint that is used to configure the foundational services IAM OAuth flow. In the Create foundational service IAM OIDC identity provider section, you use **IBM-IAM** as the name of the OIDC provider.

See the following example:

```
apiVersion: oidc.security.ibm.com/v1
kind: Client
metadata:
  name: ocpclient
  namespace: default
spec:
  secret: ocpclientsecret
  oidcLibertyClient:
    post_logout_redirect_uris:
    - 'https://console-openshift-console.apps.eying.com:443'
    redirect_uris:
    - 'https://oauth-openshift.apps.eying.com/oauth2callback/IBM-IAM'
    trusted_uri_prefixes:
    - 'https://console-openshift-console.apps.eying.com:443'
```

2. Create the OIDC **Client** resource.

```
oc create -f ocpclient.yaml
```

The command creates the resource, registers the client, and generates the **ocpclientsecret**, which includes the client ID and client secret.

If the resource is successfully created, you see the **client.oidc.security.ibm.com/ocpclient created** message.

3. Verify whether the registration is successful.

```
oc get client ocpclient -n default
```

If the registration is successful, you see the **Ready** parameter value as **True**.

```
NAME         SECRET             READY   AGE
ocpclient    ocpclientsecret    True    7s
```

4. Get the client ID and client secret values from the **ocpclientsecret**.

    1. Get the client ID.

    ```
    oc -n default get secret ocpclientsecret -o jsonpath='{.data.CLIENT_ID}' | base64 --
    decode
    ```

    2. Get the client secret.

    ```
    oc -n default get secret ocpclientsecret -o jsonpath='{.data.CLIENT_SECRET}' |
    base64 --decode
    ```

Create foundational service IAM OIDC identity provider

To create an identity provider, you must first create a Custom Resource (CR) with the identity provider specifications.

Complete these steps from the Red Hat® OpenShift® Container Platform console:

1. Log in to the Red Hat® OpenShift® Container Platform console as a user with cluster administrator access.
2. Click **Administration** > **Cluster Settings**.
3. Select the **Global Configuration** tab.
4. Scroll down to the **OAuth** configuration resource.
5. Click **OAuth**. You see the **OA cluster** page.
6. Scroll down to the **Identity Providers** section.
7. Open the **Add** drop-down list and click **OpenID Connect**. You see the **Add Identity Provider: OpenID Connect** page.
8. Specify the following values for the foundational services IAM OIDC identity provider:
   - Name: IBM-IAM
   - Client ID: Client ID that you retrieved in the Get the client ID and client secret values section.
   - Client secret: Client secret that you retrieved in the Get the client ID and client secret values section.
   - Issuer URL: `<IBM Cloud Pak foundational services console URL>/oidc/endpoint/OP`. For example, `https://cp-console.apps.eying.com/oidc/endpoint/OP`.

   In the **Claims** > **Preferred Username** section, specify `sub`. In the **More Options** section, provide the following information:
   - CA File: Browse to select the file that you created with the certificate content. See Get the issuer certificate authority (CA) root certificate.
9. Click **Add**.

The IAM OIDC identity provider is created.

You can now log in to the Red Hat® OpenShift® Container Platform console by using the foundational services access credentials. When you access the Red Hat® OpenShift® Container Platform console login page, you see the option to select **IBM-IAM** as the identity provider. When you select the option, you see the IBM Cloud Pak foundational services or the IBM Cloud Pak console login page. After you provide your credentials, you can access the Red Hat® OpenShift® Container Platform console.

Limitations

When you add a user to foundational service IAM, the foundational service IAM adds the user in OpenShift. Then, OpenShift creates an identity of the user because the user is from the foundational service IAM, which is the identity provider for OpenShift.

When you remove a user that was assigned to a team or a group, and if the user is not a member of any other team or group, the IAM service deletes the user in OpenShift. However, the user identity that OpenShift had created is not removed. This causes login failures. To remove the identity, you must manually delete the user identity in your OpenShift cluster.

Run these commands to remove the identity:

1. Get the list of users.

   ```
   oc get user
   ```

   Following is a sample output:

   ```
   NAME         UID                                    FULL NAME          IDENTITIES
   admin        c0beae06-72a6-11ea-9c4e-0a580afe0075                      openid-3:admin
   Sam        85b971ca-783e-11ea-aef7-0a580afe1468                        openid-3:Sam
   icpuser1     8e93a4cb-7919-11ea-aef7-0a580afe1468
   ```

2. Remove the user.

   ```
   oc delete user <user-name>
   ```

Example command:

```
oc delete user icpuser1
```

3. Get the user identity.

   ```
   oc get identity
   ```

   Following is a sample output:

   ```
   NAME                  IDP NAME    IDP USER NAME          USER NAME   USER UID
   openid-3:admin        openid-3    admin                  admin       c0beae06-72a6-11ea-
   9c4e-0a580afe0075
   openid-3:Sam         openid-3     Sam                    Sam        0f07f8f5-78ea-11ea-aef7-
   0a580afe1468
   openid-3:icpuser1     openid-3    icpuser1               icpuser1    f625a4cf-7918-11ea-
   9417-0a580afe106d
   ```

4. Delete the user identity.

   ```
   oc delete identity <IDP-Name:<IDP-user-name>>
   ```

   Here is an example command:

   ```
   oc delete identity openid-3:icpuser1
   ```

Configuring single sign-on

Configure single sign-on (SSO) between IBM Cloud Pak® for AIOps and your enterprise identity source.

Security Assertion Markup Language (SAML), an XML-based markup language, is an open standard for exchanging identity, authentication, and authorization information between an identity provider (your enterprise SAML server) and a service provider (your product cluster).

The identity provider issues authentication assertions along with a SAML SSO profile. The service provider receives these assertions and the profile.

The SSO flow can be summarized as follows:

1. A user attempts to access a service in IBM Cloud Pak for AIOps through a web browser.
2. IBM Cloud Pak for AIOps verifies whether an authentication token is present.
3. If no authentication token is present, IBM Cloud Pak for AIOps redirects the request for authentication to the enterprise SAML server of the user.
4. The enterprise SAML server presents a login page to the user.
5. If the user logs in successfully, the SAML server redirects the user, along with the SAML response, to IBM Cloud Pak for AIOps.
6. IBM Cloud Pak for AIOps generates an authentication token and grants access to the service that the user requested.

SSO can be configured with any Identity and Access Management (IAM) solution. You must first complete the SSO configuration in your cluster as instructed in Configuring SSO in IBM Cloud Pak for AIOps. Then, complete the SSO configuration by following the instructions that are provided by your IAM solution provider.

With SAML support, you can connect to any compatible SAML IdP (Identity Provider). The following are some IAM solution providers for configuring SSO by using SAML:

- Microsoft Active Directory

  For more information, see Configure SAML 2.0 provider settings for portals.

- Keycloak

  For more information, see Server Administration Guide.

- IBM Security Verify

For more information, see IBM Security Verify.

Configuring SSO in IBM Cloud Pak for AIOps

Metadata files are used for communication between IBM Cloud Pak for AIOps and your enterprise SAML server.

Note: If you see the error **404** while configuring SAML, check with the administrator whether the SAML is enabled for the Cloud Pak at the Identity Provider.

Prerequisites

You must configure a fully qualified domain name (FQDN) for accessing your cluster.

Configuring SSO

Note: If you are configuring SSO using SAML, you need to manually register the IdP by using Identity provider APIs in the following scenarios:

- If you are upgrading foundational services version to 3.12.0 version or the later foundational services versions and if you do not have the IdP registration.
- If you are installing foundational services version 3.12.0 or later for the first time.
- From foundational services version 3.12.0 or later, IdP registration is required regardless SAML has dependency on LDAP or not.

To verify if you have Idp registration or not, see Get IdP registration by query.

To configure SSO, complete the following sequence of steps:

1. Enable SAML.
2. Export the metadata of IBM Cloud Pak for AIOps to your enterprise SAML server. After you complete this task, a metadata file is downloaded.
3. Import the metadata sent by your enterprise SAML server.
4. Verify whether SAML was successfully configured.
5. Optional: Connect with an LDAP server and import users who might use the SSO request. For more information, see Configuring LDAP connection.

   Note:

   - You can also connect IBM Cloud Pak for AIOps with the same LDAP server that your enterprise SAML server uses for authentication.
   - If you are connecting LDAP using SAML, see SAML with LDAP dependency registration. As per your requirement, you can update the values of the schema elements, **`name`**, **`description`**, **`scim_base_path`**, **`token_attribute_mappings`**, and **`saml_ldap`**. To understand the use of schema elements, see Different schema elements.
6. Optional: You can directly connect with the SAML IdP if it is SCIM enabled. SCIM is supported to connect with the registered IdP and import the users using the IdP APIs. If you are manually registering the SAML with SCIM enabled IdP, see SAML with SCIM dependency registration. As per your requirement, you can update the values of the schema elements, **`name`**, **`description`**, **`idp_type`**, **`scim_base_path`**,**`token_attribute_mappings`**, **`scim_attribute_mappings`**, and **`config`**. To understand the use of schema elements, see Different schema elements.
7. If you are using SAML without LDAP dependency or SCIM enabled Idp, you can register the Idp by using SAML registration without any dependency. As per your requirement, you can update the values of schema elements, **`name`** , **`description`**, and **`token_attribute_mappings`**. To understand the use of schema elements, see Different schema elements.

You can use the application programming interface (API) or the command line interface (CLI) for configuring SSO in IBM Cloud Pak for AIOps.

Configuring SSO by using the APIs

For configuring SSO by using the APIs, see Single sign-on APIs.

Configuring SSO by using the CLI

Prerequisites

Install the IBM Cloud Pak CLI (cloudctl) tool.

Following commands are available to configure and manage SSO in your cluster:

- Enable Security Assertion Markup Language (SAML)
- Export metadata file
- Import metadata file
- Verify SSO configuration status
- Disable SAML

Enable SAML

Enable SSO.

```
cloudctl iam saml-enable
```

Export metadata file

When you run the command, a metadata file is downloaded from IBM Cloud Pak for AIOps and saved with the file name that you specify. You upload this file to your enterprise SAML server.

```
cloudctl iam saml-export-metadata --file <file_name>.xml
```

A sample metadata file resembles the following code:

```
<?xml version="1.0" encoding="UTF-8"?><md:EntityDescriptor
xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" \
entityID="https://mycluster.mydomain.com:8443/ibm/saml20/defaultSP"><md:SPSSODescriptor
AuthnRequestsSigned="true" \
WantAssertionsSigned="true" protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
\
<md:KeyDescriptor use="signing"><ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:X509Data> \
<ds:X509Certificate>MIID9zCCAd8CCQDIJbZgmPut9DANBgkqhkiG9w0BAQsFADBjMQswCQYDVQQGEwJVUzERMA8GA1
UE
.
.
btEmEMpzbGQy8Lb190tLeLZNW2zrBWbRmxzShn9ekS58aEbeD6PBTzWsKXsgYhZWWXw=</ds:X509Certificate> \
</ds:X509Data></ds:KeyInfo></md:KeyDescriptor><md:KeyDescriptor use="encryption"> \
<ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#"><ds:X509Data> \
<ds:X509Certificate>MIID9zCCAd8CCQDIJbZgmPut9DANBgkqhkiG9w0BAQsFADBjMQswCQYDVQQGEwJVUzERMA8GA1
UE
.
.
btEmEMpzbGQy8Lb190tLeLZNW2zrBWbRmxzShn9ekS58aEbeD6PBTzWsKXsgYhZWWXw=</ds:X509Certificate>
</ds:X509Data> \
</ds:KeyInfo></md:KeyDescriptor><md:SingleLogoutService
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" \
Location="https://mycluster.mydomain.com:8443/ibm/saml20/defaultSP/slo"/>
<md:AssertionConsumerService \
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST" \
Location="https://mycluster.mydomain.com:8443/ibm/saml20/defaultSP/acs" index="0"
isDefault="true"/>\
</md:SPSSODescriptor></md:EntityDescriptor>
```

Import metadata file

When you run the command, you upload the metadata file that you received from your enterprise SAML server to IBM Cloud Pak for AIOps.

```
cloudctl iam saml-upload-metadata --file <file_name>.xml
```

A sample metadata file resembles the following code:

```
<?xml version="1.0" encoding="UTF-8"?><md:EntityDescriptor
xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
entityID="https://w3id.alpha.sso.ibm.com/auth/sps/samlidp2/saml20">
<md:IDPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
<md:KeyDescriptor use="signing">
<KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
<X509Data>
<X509Certificate>MIIDhTCCAm2gAwIBAgIEOxmOOjANBgkqhkiG9w0BAQsFADBzMQswCQYDVQQGEwJVUz\
.
.
3YZ25IwGyzN5KK7XR1avMCk9GG0BbpjpqU29Wx3tWpqsh+Kl016Kc=</X509Certificate>
</X509Data>
</KeyInfo>
</md:KeyDescriptor>
<md:KeyDescriptor use="encryption">
<KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
<X509Data>
<X509Certificate>MIIDhTCCAm2gAwIBAgIEOxmOOjANBgkqhkiG9w0BAQsFADBzMQswCQYDVQQGEwJVUzELMAkGA\
.
.
GyzN5KK7XR1avMCk9GG0BbpjpqU29Wx3tWpqsh+Kl016Kc=</X509Certificate>
</X509Data>
</KeyInfo>
<md:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
</md:KeyDescriptor>
<md:ArtifactResolutionService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
Location="https://w3id.alpha.sso.ibm.com/auth/sps/samlidp2/saml20/soap" index="0"
isDefault="true"/>
<md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://w3id.alpha.sso.ibm.com/auth/sps/samlidp2/saml20/slo"/>
<md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://w3id.alpha.sso.ibm.com/auth/sps/samlidp2/saml20/slo"/>
<md:ManageNameIDService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://w3id.alpha.sso.ibm.com/auth/sps/samlidp2/saml20/mnids"/>
<md:ManageNameIDService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://w3id.alpha.sso.ibm.com/auth/sps/samlidp2/saml20/mnids"/>
<md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:persistent</md:NameIDFormat>
<md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</md:NameIDFormat>
<md:NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:encrypted</md:NameIDFormat>
<md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress</md:NameIDFormat>
<md:NameIDFormat>urn:oasis:names:tc:SAML:1.1:nameid-format:unspecified</md:NameIDFormat>
<md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://w3id.alpha.sso.ibm.com/auth/sps/samlidp2/saml20/login"/>
<md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://w3id.alpha.sso.ibm.com/auth/sps/samlidp2/saml20/login"/>
</md:IDPSSODescriptor>
<md:Organization>
<md:OrganizationName xml:lang="en">IBM</md:OrganizationName>
<md:OrganizationDisplayName xml:lang="en">IBM</md:OrganizationDisplayName>
<md:OrganizationURL xml:lang="en"/>
</md:Organization>
<md:ContactPerson contactType="technical">
<md:Company>IBM</md:Company>
<md:GivenName/>
<md:SurName/>
<md:EmailAddress/>
<md:TelephoneNumber/>
</md:ContactPerson>
</md:EntityDescriptor>
```

Verify SSO configuration status

Verify whether SSO is correctly configured. The command returns **true** only when SAML is enabled and the metadata file that you received from your enterprise SAML server is uploaded to IBM Cloud Pak for AIOps.

```
cloudctl iam saml-status
```

Disable SAML

Disable SSO.

```
cloudctl iam saml-disable
```

Configuring LDAP connection

Configure an LDAP (Lightweight Directory Access Protocol) connection for your IBM Cloud Pak for AIOps cluster.

You must connect an LDAP directory with your IBM Cloud Pak for AIOps cluster for managing access for users and user groups.

With an LDAP connection created, you can add users from the LDAP directory to the platform for IBM Cloud Pak for AIOps for accessing the IBM Cloud Pak Automation console. When you are adding a user from an LDAP server to IBM Cloud Pak for AIOps and are searching for a user by their name, email address, or username, the LDAP login ID for the user is considered to be the username for that user.

About this task

The following LDAP types are supported:

- IBM Tivoli Directory Server
- IBM Lotus Domino
- IBM SecureWay Directory Server
- Novell eDirectory
- Sun Java™ System Directory Server
- Netscape Directory Server
- Microsoft Active Directory
- Custom

Note: You can configure an account lockout policy when you set up your LDAP server. An account lockout policy provides more security by restricting access to the account if multiple login attempts fail.

Required user type or access level: Cluster administrator or Cloud Pak administrator

Connecting to your LDAP directory

Follow these steps to set up your LDAP connection.

1. Log on to the IBM Cloud Pak Automation console.
2. From the navigation menu, click **Administration** > **Access control**.
3. Click **Identity provider configuration**.
4. On the *Identity providers* page, click **Create connection**. The "LDAP connection" page is displayed.
5. Enter the following details to set up your LDAP connection.

   Note: You can configure multiple LDAP connection instances to the same LDAP server. However, both the **`Base DN`** and connection **`Name`** must be unique.

   - **LDAP connection** - Enter the connection information.
     - **Connection name**: A unique name for the LDAP connection. Format: 1 - 50 alphanumeric characters; Special characters that are allowed: `-` `_`
     - **Server type**: A list of directory server types to which you can connect. Select one from the list.
   - **LDAP authentication** - Enter authentication information. After you enter the information, click **Test connection** to verify whether the LDAP connection details are valid.
     - **Base DN**: The distinguished name of the search base. Example: dc=abc,dc=com. Format: 1 - 255 alphanumeric characters; Special characters that are allowed: `=` `.` `,` `-`

- **Bind DN**: The user who is allowed to search the base DN. Example: cn=admin,dc=abc,dc=com. This parameter is optional. If no user is specified in the `Bind DN` parameter, the LDAP connection is established without authentication. Format: 0 - 255 alphanumeric characters; white space is allowed; Special characters that are allowed: `= . , -`
- **Bind DN password**: The password of the user who is mentioned in the `Bind DN`. This parameter is not required if you do not specify a user in the bind DN. A maximum of 255 characters are allowed.

Note: The configuration of **Base DN** and **Bind DN** values must be set as case-sensitive and must be a full distinguished name (DN) path. The DN path, including spaces, commas, and other characters, must be the same as configured in the LDAP server. See the following example:

```
Base DN : DC=mycompany,DC=com
Bind DN : CN=Administrator,CN=Users,DC=mycompany,DC=com
```

For `Base DN`, the following values are invalid:

- dc=mycompany,dc=com because `DC` is lowercase alphabet.
- DC=mycompany, DC=com because there is a space between the parameters.

For `Bind DN`, the following values are invalid:

- cn=Administrator,cn=Users,dc=mycompany,dc=com because `CN` and `DC` are lowercase alphabets.
- CN=Administrator,DC=mycompany,DC=com because `CN=Users` parameter is missing.
- CN=Administrator,CN=Users, DC=mycompany,DC=com because there is a space between the parameters.
- CN=administrator,CN=Users,DC=mycompany,DC=com because the `administrator` parameter value starts with a lowercase alphabet.

Note: Microsoft Active Directory server does a strict check of `Base DN` and `Bind DN` values while it establishes a connection.

- **LDAP server** - Enter the server URL for your connection. After you enter the URL, click **Test connection** to verify whether the LDAP connection details are valid.
  - **URL**: The LDAP directory domain name or IP address, and the LDAP port number. The domain name must begin with `ldap(s)://`. Example URL: `ldap(s)://corpldap.abc.com:389` or `ldap://10.10.10.1:389`.

    For LDAP over SSL (LDAPS), you must use the domain name, and the URL must begin with `ldaps://`. Example URL: `ldaps://corpldap.abc.com:636`.

    Note: If you are unable to connect to your LDAPS server by using the hostname, add the IP address and host name of the LDAPS server in your local DNS. The LDAPS server hostname must be resolvable from your master node.
- **LDAP filters** - Enter information about the search filters. For default LDAP filters by LDAP type, see Default LDAP filters by LDAP type.

  Note: If your LDAP server is compliant with LDAP version 3, then the filters are not sensitive to alphabet case or to white spaces.
  - **Group filter**: The filter clause for searching groups. Format: 1 - 255 alphanumeric characters; Special characters that are allowed: white space, `= ; . , & % () {} <> |`
  - **Group ID map**: The filter to map a group name to an LDAP entry. Format: 1 - 255 alphanumeric characters; Special characters that are allowed: white space, `* : = ; . , & % () {}`
  - **Group member ID map**: The filter to map a user to a group. Format: 1 - 255 alphanumeric characters; Special characters that are allowed: white space, `* : = ; . , & % () {}`
  - **User filter**: The filter clause for searching users. Format: 1 - 255 alphanumeric characters; Special characters that are allowed: white space, `= ; . , & % () {} <> |`
  - **User ID map**: The filter to map a username to an LDAP entry. Format: 1 - 255 alphanumeric characters; Special characters that are allowed: white space, `* : = ; . , & % () {}`

6. Click **Create**.

Your cluster is now connected with your LDAP directory. Next, you can add your LDAP users and user groups to your cluster.

Notes:

- To block other hosts, see Adding egress firewall to prevent DNS loop to unknown hosts

- If you are using an LDAPS connection, the SSL (Secure Sockets Layer) certificates that are required for your LDAPS connection are automatically configured when you connect with your directory. However, you must manually restart the **auth-idp** pod. Complete these steps on your controller node:

  1. Install the OpenShift CLI (**oc**) tool.
  2. Get the **auth-idp** pods.

     ```
     oc -n ibm-common-services get pods | grep auth-idp
     ```

  3. Delete the **auth-idp** pods.

     ```
     oc -n ibm-common-services delete pods <pod_name>
     ```

  4. Wait for the pods to restart.

     If your LDAPS connection is not successful, you can try setting up the connection manually.

Default LDAP filters by LDAP type

Table 1. List of default IBM Tivoli Directory Server LDAP filters

Attribute name	Data type	Default value
groupFilter	String	(&(cn=%v)(objectclass=groupOfUniqueNames))
groupIdMap	String	*:cn
groupMemberIdMap	String	groupOfUniqueNames:uniquemember
userFilter	String	(&(emailAddress=%v)(objectclass=person))
userIdMap	String	*:emailAddress

Table 2. List of default Custom Server LDAP filters

Attribute name	Data type	Default value
groupFilter	String	(&(cn=%v)(objectclass=groupOfUniqueNames))
groupIdMap	String	*:cn
groupMemberIdMap	String	groupOfUniqueNames:uniquemember
userFilter	String	(&(uid=%v)(objectclass=ePerson))
userIdMap	String	*:uid

Table 3. List of default Microsoft Active Directory LDAP filters

Attribute name	Data type	Default value
groupFilter	String	(&(cn=%v)(objectclass=group))
groupIdMap	String	*:cn
groupMemberIdMap	String	memberOf:member
userFilter	String	(&(sAMAccountName=%v)(objectclass=user))
userIdMap	String	user:sAMAccountName

Table 4. List of default IBM Lotus Domino LDAP filters

Attribute name	Data type	Default value
groupFilter	String	(&(cn=%v)(objectclass=dominoGroup))
groupIdMap	String	*:cn
groupMemberIdMap	String	dominoGroup:member
userFilter	String	(&(uid=%v)(objectclass=Person))
userIdMap	String	person:uid

Table 5. List of default IBM SecureWay Directory Server LDAP filters

Attribute name	Data type	Default value
groupFilter	String	(&(cn=%v)((objectclass=groupOfNames)(objectclass=groupOfUniqueNames)))
groupIdMap	String	*:cn
groupMemberIdMap	String	groupOfNames:member;groupOfUniqueNames:uniqueMember
userFilter	String	(&(uid=%v)(objectclass=ePerson))
userIdMap	String	*:uid

Table 6. List of default Sun Java System Directory Server LDAP filters

Attribute name	Data type	Default value
groupFilter	String	(&(cn=%v)(objectclass=ldapsubentry))
groupIdMap	String	*:cn
groupMemberIdMap	String	nsRole:nsRole
userFilter	String	(&(uid=%v)(objectclass=inetOrgPerson))
userIdMap	String	inetOrgPerson:uid

Table 7. List of default Netscape Directory Server LDAP filters

Attribute name	Data type	Default value
groupFilter	String	(&(cn=%v)((objectclass=groupOfNames)(objectclass=groupOfUniqueNames)))
groupIdMap	String	*:cn
groupMemberIdMap	String	groupOfNames:member;groupOfUniqueNames:uniqueMember
userFilter	String	(&(uid=%v)(objectclass=inetOrgPerson))
userIdMap	String	inetOrgPerson:uid

Table 8. List of default Novell eDirectory LDAP filters

Attribute name	Data type	Default value
groupFilter	String	(&(cn=%v)(objectclass=groupOfNames))
groupIdMap	String	*:cn
groupMemberIdMap	String	groupOfNames:member
userFilter	String	(&(cn=%v)(objectclass=Person))
userIdMap	String	person:cn

Adding egress firewall to prevent DNS loop to unknown hosts

Run the following command to retrieve the egress network policy:

```
oc get egressnetworkpolicy ldap-firewall
```

Sample output:

```
NAME             AGE
ldap-firewall    16m
```

The following sample YAML spec defines the LDAP firewall egress network policy. To block other hosts, update the spec to specify the hosts that you want to deny access. Then, save and apply the updated policy.

```
apiVersion: network.openshift.io/v1
kind: EgressNetworkPolicy
metadata:
  name: ldap-firewall
spec:
  egress:
  - type: Allow
    to:
      dnsName: bluepages.ibm.com
  - type: Allow
    to:
      cidrSelector:  9.30.192.0/24
  - type: Deny
    to:
      cidrSelector: 0.0.0.0/0
  podSelector:
    matchLabels:
      app: auth-idp
  policyType:
  - Egress
```

What to do next

Now, you can add LDAP users and user groups to your clusters and assign permissions.

- Add users
- Add user groups

Changing LDAP cache settings

Changing the Lightweight Directory Access Protocol (LDAP) cache settings that are used for authentication in IBM Cloud Pak® for AIOps.

The LDAP cache setting parameters are listed in the following tables. For more information, see LDAP User Registry (ldapRegistry).

Note: Specify a positive integer followed by a unit of time, which can be hours (h), minutes (m), seconds (s), or milliseconds (ms). For example, specify 500 milliseconds as 500ms. You can include multiple values in a single entry. For example, 1s500ms is equivalent to 1.5 seconds. This note is applicable only to the `LDAP_ATTR_CACHE_TIMEOUT` and `LDAP_SEARCH_CACHE_TIMEOUT` parameters.

Table 1. LDAP attribute cache properties

Parameter	Description	Default value
`LDAP_ATTR_CACHE_ENABLED`	Enable or disable LDAP attribute cache.	true
`LDAP_ATTR_CACHE_SIZE`	Number of entities that can be stored in the cache.	2000
`LDAP_ATTR_CACHE_SIZELIMIT`	Maximum number of attributes per LDAP entity that are cached.	2000
`LDAP_ATTR_CACHE_TIMEOUT`	Maximum time that the contents of the LDAP attribute cache are available. When the specified time elapses, the LDAP attribute cache is cleared.	1200s

Table 2. LDAP search results cache properties

Parameter	Description	Default value
`LDAP_SEARCH_CACHE_ENABLED`	Enable or disable LDAP search results cache.	true
`LDAP_SEARCH_CACHE_SIZE`	Number of search results that are stored in the cache.	2000
`LDAP_SEARCH_CACHE_SIZELIMIT`	Maximum number of results that can be cached for a single LDAP search.	2000
`LDAP_SEARCH_CACHE_TIMEOUT`	Maximum time that the contents of the search results cache are available. When the specified time elapses, the search results cache is cleared.	1200s

Changing the parameter values by using kubectl

To change the parameter values, complete the following steps:

1. Install the Kubernetes CLI (kubectl) tool.
2. Edit the `platform-auth-idp` ConfigMap.

   ```
   kubectl -n ibm-common-services edit configmap platform-auth-idp
   ```

3. Change the following attribute values as required:
   - `LDAP_ATTR_CACHE_ENABLED`
   - `LDAP_ATTR_CACHE_SIZE`
   - `LDAP_ATTR_CACHE_SIZELIMIT`
   - `LDAP_ATTR_CACHE_TIMEOUT`
   - `LDAP_SEARCH_CACHE_ENABLED`
   - `LDAP_SEARCH_CACHE_SIZE`
   - `LDAP_SEARCH_CACHE_SIZELIMIT`
   - `LDAP_SEARCH_CACHE_TIMEOUT`
4. Save and close the ConfigMap.

5. Restart the **auth-idp** pods

   ```
   kubectl -n ibm-common-services delete pod -l k8s-app=auth-idp
   ```

6. Wait for some time. Then, check the status of the **auth-idp** pods. The status must show as **4/4 Running** for all the pods.

   ```
   kubectl -n ibm-common-services get pods | grep auth-idp
   ```

Changing the parameter values by using the console

1. Log in to the Red Hat® OpenShift® Container Platform console as a user with cluster administrator access.
2. From the navigation menu, click **Workloads** > **Config Maps.**
3. Search for **platform-auth-idp**.
4. Click **...** > **Edit Config Map**.
5. Change the following attribute values as required:
   - **LDAP_ATTR_CACHE_ENABLED**
   - **LDAP_ATTR_CACHE_SIZE**
   - **LDAP_ATTR_CACHE_SIZELIMIT**
   - **LDAP_ATTR_CACHE_TIMEOUT**
   - **LDAP_SEARCH_CACHE_ENABLED**
   - **LDAP_SEARCH_CACHE_SIZE**
   - **LDAP_SEARCH_CACHE_SIZELIMIT**
   - **LDAP_SEARCH_CACHE_TIMEOUT**
6. Click **Save**.
7. From the navigation menu, click **Workloads > Deployments**.
8. Locate **auth-idp**.
9. Click **...** > **Edit Deployment**. A window for editing displays.
10. Click **Save** without making any change. This step is to reload the **auth-idp** pods with the latest ConfigMap values.
11. Click **auth-idp**.
12. Wait for some time. Then, check the status of the **auth-idp** pods in the **Pods** pane. The status of all the pods must show as **4/4** under the **Ready** field name.

Changing LDAP search settings

Changing the Lightweight Directory Access Protocol (LDAP) search settings in IBM Cloud Pak® for AIOps.

Configure attributes consideration for LDAP search

By default, IBM Cloud Pak for AIOps searches for users and groups on both the common name (CN) and identifier (ID) attributes as configured in the LDAP connection settings. Examples of ID attributes are **uid**, **sAMAccountName**, and **emailAddress**.

This behavior can be changed by setting the following configuration parameters:

The following two attributes are used for the search:

- **LDAP_SEARCH_CN_ATTR_ONLY** - Search by CN only. Default value is **false**.
- **LDAP_SEARCH_ID_ATTR_ONLY**- Search by ID only. Default value is **false**.

To use only the CN attribute for searching users or groups, set **LDAP_SEARCH_CN_ATTR_ONLY** value to **true**. To use only the ID attribute for searching users or groups, set **LDAP_SEARCH_ID_ATTR_ONLY** value to **true**.

Following are the steps to change the values of the LDAP search config parameters:

Changing the attribute values by using kubectl

To change the values, complete the following steps:

1. Install the **kubectl** CLI.
2. Edit the **platform-auth-idp** ConfigMap.

   ```
   kubectl -n ibm-common-services edit configmap platform-auth-idp
   ```

   Set the following attribute values as required to **true** or **false**.

   - **LDAP_SEARCH_CN_ATTR_ONLY**
   - **LDAP_SEARCH_ID_ATTR_ONLY**

   Note: Set only one value to **true**.
3. Save and close the ConfigMap.
4. Restart the **auth-idp** pods

   ```
   kubectl -n ibm-common-services delete pod -l k8s-app=auth-idp
   ```
5. Wait for some time. Then, check the status of the **auth-idp** pods. The status must show as **4/4 Running** for all the pods.

   ```
   kubectl -n ibm-common-services get pods | grep auth-idp
   ```

Changing the attribute values by using the console

1. Log in to the Red Hat® OpenShift® Container Platform console as a user with cluster administrator access.
2. From the navigation menu, click **Workloads** > **Config Maps.**
3. Search for **platform-auth-idp**.
4. Click **...** > **Edit Config Map**.
5. Change either of the following attribute values to **true**, as required:

   - **LDAP_SEARCH_CN_ATTR_ONLY**
   - **LDAP_SEARCH_ID_ATTR_ONLY**

     Note: Set only one value to **true**.
6. Click **Save**.
7. From the navigation menu, click **Workloads > Deployments**.
8. Locate **auth-idp**.
9. Click **...** > **Edit Deployment**. A window for editing displays.
10. Click **Save** without making any change. This step is to reload the **auth-idp** pods with the latest ConfigMap values.
11. Click **auth-idp**.
12. Wait for some time. Then, check the status of the **auth-idp** pods in the **Pods** pane. The status of all the pods must show as **4/4** under the **Ready** field name.

Changing the LDAP search size limit and time limit values

Following are the LDAP search limit configuration parameters:

- **`LDAP_SEARCH_SIZE_LIMIT`**: Default value is "50". Range of values is 50 - 100.
- **`LDAP_SEARCH_TIME_LIMIT`**: Default value is "5". Range of values is 5 - 50.

Changing the values by using kubectl

To change the values, complete the following steps:

1. Set up the **`kubectl`** CLI. For more information, see Installing the Kubernetes CLI (kubectl).
2. Edit the **`platform-auth-idp`** ConfigMap.

   ```
   kubectl -n ibm-common-services edit configmap platform-auth-idp
   ```

   Change the following values as required:

   - **`LDAP_SEARCH_SIZE_LIMIT`**
   - **`LDAP_SEARCH_TIME_LIMIT`**
3. Save and close the ConfigMap.
4. Restart the **`auth-idp`** pods

   ```
   kubectl -n ibm-common-services delete pod -l k8s-app=auth-idp
   ```
5. Wait for some time. Then, check the status of the **`auth-idp`** pods. The status must show as **`4/4 Running`** for all the pods.

   ```
   kubectl -n ibm-common-services get pods | grep auth-idp
   ```

Changing the values by using the console

1. Log in to the console as a user with cluster administrator access.
2. From the navigation menu, click **Workloads** > **Config Maps.**
3. Search for **`platform-auth-idp`**.
4. Click **...** > **Edit Config Map**.
5. Change the following attribute values, as required:

   - **`LDAP_SEARCH_SIZE_LIMIT`**
   - **`LDAP_SEARCH_TIME_LIMIT`**
6. Click **Submit**.
7. From the navigation menu, click **Workloads > Deployments**.
8. Locate **`auth-idp`**.
9. Click **...** > **Edit Deployment**. A window for editing displays.
10. Click **Submit** without making any change. This step is to reload the **`auth-idp`** pods with the latest ConfigMap values.
11. Click **`auth-idp`**.
12. Wait for some time. Then, check the status of the **`auth-idp`** pods in the **Pods** pane. The status of all the pods must show as **`4/4`** under the **Ready** field name.

Change LDAP search wild characters usage

Following is the LDAP search wild characters configuration parameter:

- **`LDAP_SEARCH_EXCLUDE_WILDCARD_CHARS`**: Default value is "false".

In IBM Cloud Pak for AIOps, wild characters such as asterisk are used in the search string to match the substring value of the LDAP server attribute.

Set **LDAP_SEARCH_EXCLUDE_WILDCARD_CHARS** value to **true** to exclude wild characters (*) in the search string and to match the exact value of the LDAP server attribute.

Following is the default search string:

```
(|(&(cn=*<searchstring>*)(objectclass=person))(&(uid=*<searchstring>*)(objectclass=person)))
```

Following is the search string when the **LDAP_SEARCH_EXCLUDE_WILDCARD_CHARS** is set to **true**:

```
(|(&(cn=<searchstring>)(objectclass=person))(&(uid=<searchstring>)(objectclass=person)))
```

To change the wild characters usage in the search string, complete the following steps:

Changing the wild characters configuration parameter by using kubectl

To change the values, complete the following steps:

1. Install the Kubernetes CLI (kubectl) tool.
2. Edit the **platform-auth-idp** ConfigMap.

   ```
   kubectl -n ibm-common-services edit configmap platform-auth-idp
   ```

   Change the following value as required:

   - **LDAP_SEARCH_EXCLUDE_WILDCARD_CHARS**
3. Save and close the ConfigMap.
4. Restart the **auth-idp** pods

   ```
   kubectl -n ibm-common-services delete pod -l k8s-app=auth-idp
   ```
5. Wait for some time. Then, check the status of the **auth-idp** pods. The status must show as **4/4 Running** for all the pods.

   ```
   kubectl -n ibm-common-services get pods | grep auth-idp
   ```

Changing the wild characters configuration parameter by using the console

1. Log in to the Red Hat® OpenShift® Container Platform console as a user with cluster administrator access.
2. From the navigation menu, click **Configuration** > **Config Maps.**
3. Search for **platform-auth-idp**.
4. Click **...** > **Edit Config Map**.
5. Change the following value, as required:
   - **LDAP_SEARCH_EXCLUDE_WILDCARD_CHARS**
6. Click **Save**.
7. From the navigation menu, click **Workloads > Deployments**.
8. Locate **auth-idp**.
9. Click **...** > **Edit**. A window for editing displays.
10. Click **Save** without making any change. This step is to reload the **auth-idp** pods with the latest ConfigMap values.
11. Click **auth-idp**.
12. Wait for some time. Then, check the status of the **auth-idp** pods in the *Pods* pane. The status of all the pods must show as **4/4** under the **Ready** field name.

Changing LDAP search cache variable values

Change the default values of the Lightweight Directory Access Protocol (LDAP) search cache variables.

You can change the values of the following variables:

Table 1. LDAP cache parameters

Variable	Default value	Description

Variable	Default value	Description
`LDAP_SEARCH_CACHE_TIMEOUT`	1200 seconds	The maximum time that the contents of the search results cache are available.
`LDAP_SEARCH_CACHE_ENABLED`	true	A Boolean value to indicate that the search results must be cached or not.
`LDAP_SEARCH_CACHE_SIZE`	2000	The number of search results that are stored in the cache.
`LDAP_SEARCH_CACHE_SIZELIMIT`	2000	The maximum number of results that can be cached for a single LDAP search.
`LDAP_ATTR_CACHE_SIZE`	2000	The number of entities that can be stored in the cache.
`LDAP_ATTR_CACHE_TIMEOUT`	1200 seconds	The maximum time that the contents of the LDAP attribute cache are available.
`LDAP_ATTR_CACHE_SIZELIMIT`	2000	The maximum number of attributes per LDAP entity that are cached.
`LDAP_ATTR_CACHE_ENABLED`	true	A Boolean value to indicate that the entities must be cached or not.

For more information, see LDAP User Registry (ldapRegistry).

Follow these steps to change the variable values:

1. Install the Kubernetes CLI (kubectl).
2. Edit the **platform-auth-idp** configmap.

   ```
   kubectl -n ibm-common-services edit configmap platform-auth-idp
   ```

   The file content resembles the following code:

   ```
   # Please edit the following object. Lines beginning with a '#' will be ignored,
   # and an empty file will abort the edit. If an error occurs while saving this file will
   be
   # reopened with the relevant failures.
   #
   apiVersion: v1
   data:
     AUDIT_DETAIL: "false"
     AUDIT_ENABLED_IDMGMT: "false"
     AUDIT_ENABLED_IDPROVIDER: "false"
     .
     .
     LDAP_ATTR_CACHE_ENABLED: "true"
     LDAP_ATTR_CACHE_SIZE: "2000"
     LDAP_ATTR_CACHE_SIZELIMIT: "2000"
     LDAP_ATTR_CACHE_TIMEOUT: 1200s
     LDAP_SEARCH_CACHE_ENABLED: "true"
     LDAP_SEARCH_CACHE_SIZE: "2000"
     LDAP_SEARCH_CACHE_SIZELIMIT: "2000"
     LDAP_SEARCH_CACHE_TIMEOUT: 1200s
     LDAP_SEARCH_CN_ATTR_ONLY: "false"
     LDAP_SEARCH_ID_ATTR_ONLY: "false"
     LDAP_SEARCH_SIZE_LIMIT: "50"
     LDAP_SEARCH_TIME_LIMIT: "5"
     MASTER_HOST: mycluster.icp
     NODE_ENV: production
     .
     .
   "/tmp/kubectl-edit-i5ta2.yaml" 60L, 2204C
   ```

3. Change the LDAP variable values, as required, and save the ConfigMap.

   After you update and save the changes, you see the following message:

   ```
   configmap "platform-auth-idp" edited
   ```

4. Restart the **auth-idp** pod.

   1. Get the **auth-idp** pod ID.

      ```
      kubectl -n ibm-common-services get pods | grep auth-idp
      ```

Following is a sample output of the command:

```
auth-idp-5b78f                          4/4     Running                 3          12d
```

2. Delete the **auth-idp** pod.

   ```
   kubectl -n ibm-common-services delete pod auth-idp-5b78f
   ```

   After the pod is deleted, you see the following message:

   ```
   pod "auth-idp-5b78f" deleted
   ```

   Wait for a few minutes for the pod to restart.

3. Verify the status of the pod.

   ```
   kubectl -n ibm-common-services get pods | grep auth-idp
   ```

   The pod gets a new ID after restart.

   ```
   auth-idp-5267t                              1/4       Running     0          4m
   ```

5. Verify that the LDAP variables are updated.

   1. Get the **platform-auth** container ID.

      ```
      docker ps | grep platform-auth
      ```

      Following is a sample output:

      ```
      2e5d416fe6d8        3ea9fbf2c199                            "/usr/bin/supervisor…"
      12 days ago         Up 12 days                                k8s_platform-auth-
      service_auth-idp-5b78f_kube-system_b83319c3-2484-11e9-8d6a-00000a29093b_1
      ```

   2. Log in to the **platform-auth** container.

      ```
      docker exec -it <container ID> bash
      ```

      Following is a sample command and output:

      ```
      docker exec -it 2e5d416fe6d8 bash
      bash-4.4#
      ```

   3. List the LDAP variables.

      ```
      env|grep LDAP
      ```

      You can see the updated value in the output.

      ```
      LDAP_ATTR_CACHE_SIZE=2500
      LDAP_ATTR_CACHE_ENABLED=true
      LDAP_ATTR_CACHE_TIMEOUT=1200s
      LDAP_ATTR_CACHE_SIZELIMIT=2000
      LDAP_SEARCH_CACHE_ENABLED=true
      LDAP_SEARCH_CACHE_SIZE=2000
      LDAP_SEARCH_CACHE_SIZELIMIT=2000
      LDAP_SEARCH_CACHE_TIMEOUT=1200s
      ```

Updating SCIM LDAP attributes mapping

You can update the SCIM LDAP attributes mapping.

To support the SCIM APIs for a configured LDAP connection in IBM Cloud Pak foundational services, you can update **SCIM_LDAP_ATTRIBUTES_MAPPING** data in the following ways:

- By using SCIM configuration for LDAP connection. For more information, see SCIM configuration by using IBM Cloud Pak for AIOps UI.
- By using the **attributemapping** API (available for IBM Cloud Pak foundational services version 3.11.0 or later)

- By using the `platform-auth-idp` configmap

Note: Use of the configmap is deprecated from the IBM Cloud Pak foundational services version 3.17.0 and is subject to be removed in the future release.

Updating with `attributemapping` API

To use the **`attributemapping`** API, you must add an authorization header to your request. You need to add an access token to the authorization header. To obtain the access token, see Preparing to run component or management API commands.

Procedure

Note: The **`members`** and **`objectClass`** attributes do not consider the custom SCIM attribute values. Use LDAP filter values for these attributes to map the data.

1. Check the existing attribute mappings by using the following API call:

   ```
   curl -sk -X GET --header "Authorization: Bearer $ACCESS_TOKEN" --header "Content-Type:
   application/json" "https://$BCS_URL:443/idmgmt/identity/api/v1/scim/attributemappings"
   ```

   The response resembles the following code:

   ```
   [{"idp_id":"default","idp_type":"ldap","user":
   {"id":"dn","userName":"uid","principalName":"uid","displayName":"cn","givenName":"cn","fa
   milyName":"sn","fullName":"cn","externalId":"dn","emails":"mail","created":"createTimesta
   mp","lastModified":"modifyTimestamp","phoneNumbers":[{"value":"mobile","type":"mobile"},
   {"value":"telephoneNumber","type":"work"}],"objectClass":"person","groups":"memberOf"},"g
   roup":
   {"id":"dn","name":"cn","principalName":"cn","displayName":"cn","externalId":"dn","created
   ":"createTimestamp","lastModified":"modifyTimestamp","objectClass":"groupOfUniqueNames","
   members":"uniqueMember"}},
   ```

2. Create the attribute mappings with your mapping data by using the following API call:

   Note: If you already have the attribute mappings, do not create another attribute mapping. You can skip this step.

   ```
   export DATA='{"idp_id":"test","dip_type":"ldap","user":
   {"id_test":"dn","userName":"uid","principalName":"uid","displayName":"cn","givenName":"cn
   ","familyName":"sn","fullName":"cn","externalId":"dn","phoneNumbers":
   [{"value":"mobile","type":"mobile"},
   {"value":"telephoneNumber","type":"work"}],"objectClass":"person","groups":"memberOf"},"g
   roup":
   {"id":"dn","name":"cn","principalName":"cn","displayName":"cn","externalId":"dn","created
   ":"createTimestamp","lastModified":"modifyTimestamp","objectClass":"groupOfUniqueNames","
   members":"uniqueMember"}}'
   ```

   The sample curl command resembles the following code:

   ```
   curl -sk -X POST --header "Authorization: Bearer $ACCESS_TOKEN" --header "Content-Type:
   application/json" -d $DATA
   "https://$BCS_URL:443/idmgmt/identity/api/v1/scim/attributemappings"
   ```

   The response resembles the following code:

   ```
   {"idp_id":"test","idp_type":"ldap","user":
   {"id_test":"dn","userName":"uid","principalName":"uid","displayName":"cn","givenName":"cn
   ","familyName":"sn","fullName":"cn","externalId":"dn","phoneNumbers":
   [{"value":"mobile","type":"mobile"},
   {"value":"telephoneNumber","type":"work"}],"objectClass":"person","groups":"memberOf"},"g
   roup":
   {"id":"dn","name":"cn","principalName":"cn","displayName":"cn","externalId":"dn","created
   ":"createTimestamp","lastModified":"modifyTimestamp","objectClass":"groupOfUniqueNames","
   members":"uniqueMember"}}
   ```

3. Update the existing attribute mappings data with your mapping data by using the following API call:

   ```
   export DATA='{"idp_id":"$IDP_ID","idp_type":"ldap","user":
   {"id_test":"dn","userName":"uid","principalName":"uid","displayName":"cn","givenName":"cn
   ","familyName":"sn","fullName":"cn","externalId":"dn","phoneNumbers":
   [{"value":"mobile","type":"mobile"},
   ```

```
{"value":"telephoneNumber","type":"work"}],"objectClass":"person","groups":"memberOf"}},"
group":
{"id":"dn","name":"cn","principalName":"cn","displayName":"cn","externalId":"dn","created
":"createTimestamp","lastModified":"modifyTimestamp","objectClass":"groupOfUniqueNames","
members":"uniqueMember"}'
```

The sample curl command resembles the following code:

```
curl -sk -X PUT --header "Authorization: Bearer $ACCESS_TOKEN" --header "Content-Type:
application/json" -d $DATA
"https://$BCS_URL:443/idmgmt/identity/api/v1/scim/attributemappings/$IDP_ID"
```

The response resembles the following code:

```
{"idp_id":"test","idp_type":"ldap","user":
{"id_test":"dn","userName":"uid","principalName":"uid","displayName":"cn","givenName":"cn
","familyName":"sn","fullName":"cn","externalId":"dn","phoneNumbers":
[{"value":"mobile","type":"mobile"},
{"value":"telephoneNumber","type":"work"}],"objectClass":"person","groups":"memberOf"},"g
roup":
{"id":"dn","name":"cn","principalName":"cn","displayName":"cn","externalId":"dn","created
":"createTimestamp","lastModified":"modifyTimestamp","objectClass":"groupOfUniqueNames","
members":"uniqueMember"}}
```

4. Delete the existing attribute mappings by using the following API call:

```
curl -sk -X DELETE --header "Authorization: Bearer $ACCESS_TOKEN" \
--header "Content-Type: application/json" \
"https://$BCS_URL:443/idmgmt/identity/api/v1/scim/attributemappings/$IDP_ID"
```

The response resembles the following code:

```
{"count":1}
```

Updating with `platform-auth-idp` configmap

You can update **`SCIM_LDAP_ATTRIBUTES_MAPPING`** data in the configmap **`platform-auth-idp`** to support SCIM APIs for a configured LDAP connection in IBM Cloud Pak foundational services.

Currently, the **`platform-auth-idp`** configmap contains the following default configuration. Within this configuration, **`default`** is the default name. This configuration also has two sets of mapping data for each **`user`** and **`group`** SCIM resource.

In each data set, **`key`** is the SCIM attribute name and **`value`** is the LDAP attribute name, for example, **`"userName": "uid"`**. See the following parameter description:

- **`userName`** is the SCIM attribute name.
- **`uid`** is the LDAP attribute name.

Default configuration

```
{
    "default": {
        "user": {
            "id": "dn",
            "userName": "uid",
            "principalName": "uid",
            "displayName": "cn",
            "givenName": "cn",
            "familyName": "sn",
            "fullName": "cn",
            "externalId": "dn",
            "emails": "mail",
            "created": "createTimestamp",
            "lastModified": "modifyTimestamp",
            "phoneNumbers": [{
                    "value": "mobile",
                    "type": "mobile"
                },
                {
                    "value": "telephoneNumber",
```

```
                    "type": "work"
                }],
            "objectClass": "person",
            "groups": "memberOf"
        },
        "group": {
            "id": "dn",
            "name": "cn",
            "principalName": "cn",
            "displayName": "cn",
            "externalId": "dn",
            "created": "createTimestamp",
            "lastModified": "modifyTimestamp",
            "objectClass": "groupOfUniqueNames",
            "members": "uniqueMember"
        }
    }
}
```

You can customize the configuration to meet your requirements for the configured LDAP connection. Then, you can construct a similar mapping for your LDAP server and append the mapping to the **SCIM_LDAP_ATTRIBUTES_MAPPING** data in the **platform-auth-idp** configmap.

As an example, you can use **Connection name** as **pmsad** in your configured MSAD LDAP connection configuration. Next, you can use **pmsad** as the configuration key and construct the mapping data as shown in the following code example. Then, you can append the mapping data to your configmap data. For more information about attributes, see LDAP attributes.

```
    "pmsad": {
        "user": {
            "id": "dn",
            "userName": "sAMAccountName",
            "principalName": "sAMAccountName",
            "displayName": "displayName",
            "givenName": "givenName",
            "familyName": "sn",
            "fullName": "cn",
            "externalId": "dn",
            "emails": "mail",
            "created": "whenCreated",
            "lastModified": "whenChanged",
            "phoneNumbers": [{
                    "value": "mobile",
                    "type": "mobile"
                },
                {
                    "value": "telephoneNumber",
                    "type": "work"
                }],
            "objectClass": "person",
            "groups": "memberOf"
        },
        "group": {
            "id": "dn",
            "name": "cn",
            "principalName": "cn",
            "displayName": "cn",
            "externalId": "dn",
            "created": "whenCreated",
            "lastModified": "whenChanged",
            "objectClass": "group",
            "members": "member"
        }
    }
```

Procedure

Update your configmap by using the CLI. Run these commands:

1. Edit **platform-auth-idp** configmap.

```
oc -n ibm-common-services edit configmap platform-auth-idp
```

Go to the **SCIM_LDAP_ATTRIBUTES_MAPPING** attribute value and append your mapping data to the existing value. Make sure to add a comma(,) for previous entry in the JSON.

For IBM Cloud Pak foundational services version 3.11.0, set the **ATTR_MAPPING_FROM_CONFIG** attribute value to **true**.

Save and quit.

2. Check the updated data in your **platform-auth-idp** configmap. The output resembles the following code.

```
oc -n ibm-common-services get configmap platform-auth-idp -o yaml | grep -A50
SCIM_LDAP_ATTRIBUTES_MAPPING
```

```
...
{
    "default": {
        "user": {
            "id": "dn",
            "userName": "uid",
            "principalName": "uid",
            "displayName": "cn",
            "givenName": "cn",
            "familyName": "sn",
            "fullName": "cn",
            "externalId": "dn",
            "emails": "mail",
            "created": "createTimestamp",
            "lastModified": "modifyTimestamp",
            "phoneNumbers": [{
                    "value": "mobile",
                    "type": "mobile"
                },
                {
                    "value": "telephoneNumber",
                    "type": "work"
                }],
            "objectClass": "person",
            "groups": "memberOf"
        },
        "group": {
            "id": "dn",
            "name": "cn",
            "principalName": "cn",
            "displayName": "cn",
            "externalId": "dn",
            "created": "createTimestamp",
            "lastModified": "modifyTimestamp",
            "objectClass": "groupOfUniqueNames",
            "members": "uniqueMember"
        }
    },
    "pmsad": {
        "user": {
            "id": "dn",
            "userName": "sAMAccountName",
            "principalName": "sAMAccountName",
            "displayName": "displayName",
            "givenName": "givenName",
            "familyName": "sn",
            "fullName": "cn",
            "externalId": "dn",
            "emails": "mail",
            "created": "whenCreated",
            "lastModified": "whenChanged",
            "phoneNumbers": [{
                    "value": "mobile",
                    "type": "mobile"
                },
                {
                    "value": "telephoneNumber",
```

```
                    "type": "work"
                }],
            "objectClass": "person",
            "groups": "memberOf"
        },
        "group": {
            "id": "dn",
            "name": "cn",
            "principalName": "cn",
            "displayName": "cn",
            "externalId": "dn",
            "created": "whenCreated",
            "lastModified": "whenChanged",
            "objectClass": "group",
            "members": "member"
        }
    }
}
```

Note: Make sure that complete data is a valid JSON.

3. Restart the **auth-idp** pods

```
oc -n ibm-common-services delete pod -l k8s-app=auth-idp
```

4. Wait for some time. Then, check the status of the **auth-idp** pods. The status must show as **4/4 Running** for all pods.

```
oc -n ibm-common-services get pods | grep auth-idp
```

Note: By default, the LDAP connection configuration considers the following mapping values for SCIM APIs.

Table. LDAP connection attribute mappings

LDAP connection attributes	SCIM object	SCIM attributes
LDAP_USERIDMAP	user	userName
LDAP_USERFILTER	user	objectClass
LDAP_GROUPIDMAP	group	name
LDAP_GROUPFILTER	group	objectClass
LDAP_GROUPMEMBERIDMAP	group	members

Note: If you have multiple expressions in the LDAP connection attributes configuration, the first configuration value is considered.

LDAP attributes

For information about LDAP attributes, you can use the **ldapsearch** tool or any LDAP browser such as **Apache Directory Studio**.

Installing the ldapsearch tool

On Ubuntu, run the following command:

```
sudo apt-get install ldap-utils
```

On Red Hat® Enterprise Linux®, run the following command:

```
sudo yum install openldap-clients
```

The following examples show the ldapsearch tool command structure and an example output.

User

Command:

```
ldapsearch -x -H <LDAP_URL> -b <LDAP_BASEDN> -D <LDAP_BINDDN> -W -s sub "
(sAMAccountName=pavann)" "*" "+"
```

Output:

```
# extended LDIF
#
# LDAPv3
# base <DC=ibmtest,DC=com> with scope subtree
# filter: (sAMAccountName=pava)
# requesting: * +
#

# Tom Northwood, Users, ibmtest.com
dn: CN=Tom Northwood,CN=Users,DC=ibmtest,DC=com
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: user
cn: Tom Northwood
givenName: Tom Northwood
distinguishedName: CN=Tom Northwood,CN=Users,DC=ibmtest,DC=com
instanceType: 4
whenCreated: 20180802104118.0Z
whenChanged: 20210305115346.0Z
displayName: Tom Northwood
uSNCreated: 13356
memberOf: CN=group9,CN=Users,DC=ibmtest,DC=com
memberOf: CN=group8,CN=Users,DC=ibmtest,DC=com
memberOf: CN=group7,CN=Users,DC=ibmtest,DC=com
memberOf: CN=group6,CN=Users,DC=ibmtest,DC=com
memberOf: CN=group5,CN=Users,DC=ibmtest,DC=com
memberOf: CN=group4,CN=Users,DC=ibmtest,DC=com
memberOf: CN=group3,CN=Users,DC=ibmtest,DC=com
memberOf: CN=group2,CN=Users,DC=ibmtest,DC=com
memberOf: CN=group1,CN=Users,DC=ibmtest,DC=com
memberOf: CN=security,CN=Users,DC=ibmtest,DC=com
uSNChanged: 145644
name: Tom Northwood
objectGUID:: pVKE4qv5MEyqxjQ3nUvsWA==
userAccountControl: 66048
badPwdCount: 0
codePage: 0
countryCode: 0
badPasswordTime: 132285264754362226
lastLogoff: 0
lastLogon: 132285264944206361
pwdLastSet: 131776804917936927
primaryGroupID: 513
objectSid:: AQUAAAAAAAUVAAAAr/FodBO+7uVBDhlaXAQAAA==
accountExpires: 9223372036854775807
logonCount: 0
sAMAccountName: pavann
sAMAccountType: 805306368
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=ibmtest,DC=com
dSCorePropagationData: 20180802104118.0Z
dSCorePropagationData: 16010101000000.0Z
lastLogonTimestamp: 132594188266767176

# search reference
ref: ldap://ForestDnsZones.ibmtest.com/DC=ForestDnsZones,DC=ibmtest,DC=com

# search reference
ref: ldap://DomainDnsZones.ibmtest.com/DC=DomainDnsZones,DC=ibmtest,DC=com

# search reference
ref: ldap://ibmtest.com/CN=Configuration,DC=ibmtest,DC=com

# search result
search: 2
result: 0 Success

# numResponses: 5
# numEntries: 1
# numReferences: 3
```

Group

Command:

```
ldapsearch -x -H <LDAP_URL> -b <LDAP_BASEDN> -D <LDAP_BINDDN> -W -s sub "(cn=security)" "*"
"+"
```

Output:

```
# extended LDIF
#
# LDAPv3
# base <DC=ibmtest,DC=com> with scope subtree
# filter: (cn=security)
# requesting: * +
#

# security, Users, ibmtest.com
dn: CN=security,CN=Users,DC=ibmtest,DC=com
objectClass: top
objectClass: group
cn: security
member: CN=Neil Wilson,CN=Users,DC=ibmtest,DC=com
member: CN=John Winston,CN=Users,DC=ibmtest,DC=com
member: CN=Jeff Rodrigue,CN=Users,DC=ibmtest,DC=com
member: CN=Kane Shatner,CN=Users,DC=ibmtest,DC=com
member: CN=Anna Reynolds,CN=Users,DC=ibmtest,DC=com
member: CN=Gessie Lemson,CN=Users,DC=ibmtest,DC=com
member: CN=Lauri Williamson,CN=Users,DC=ibmtest,DC=com
member: CN=Tom Northwood,CN=Users,DC=ibmtest,DC=com
distinguishedName: CN=security,CN=Users,DC=ibmtest,DC=com
instanceType: 4
whenCreated: 20180802103445.0Z
whenChanged: 20180802105558.0Z
uSNCreated: 13330
memberOf: CN=isl,CN=Users,DC=ibmtest,DC=com
uSNChanged: 13455
name: security
objectGUID:: UPoQ3uNfVk+fHn1W1b5KUg==
objectSid:: AQUAAAAAAAUVAAAAr/FodBO+7uVBDhlaWAQAAA==
sAMAccountName: security
sAMAccountType: 268435456
groupType: -2147483646
objectCategory: CN=Group,CN=Schema,CN=Configuration,DC=ibmtest,DC=com
dSCorePropagationData: 16010101000000.0Z

# search reference
ref: ldap://ForestDnsZones.ibmtest.com/DC=ForestDnsZones,DC=ibmtest,DC=com

# search reference
ref: ldap://DomainDnsZones.ibmtest.com/DC=DomainDnsZones,DC=ibmtest,DC=com

# search reference
ref: ldap://ibmtest.com/CN=Configuration,DC=ibmtest,DC=com

# search result
search: 2
result: 0 Success

# numResponses: 5
# numEntries: 1
# numReferences: 3
```

IBM Cloud attributes

To search IBM Cloud users in SCIM, custom attributes can be defined as shown in the following code example. By default, these attributes are not defined.

```
"IBMCloud" = {
    "user": {
        "id": "userId",
        "userName": "userId",
        "principalName": "userId",
```

```
        "displayName": "firstName",
        "emails": "email",
        "familyName": "lastName",
        "givenName": "firstName",
        "externalId": "userId"
    }
};
```

Changing Logjam configuration property

The Logjam configuration parameter is **LOGJAM_DHKEYSIZE_2048_BITS_ENABLED**. The default value is **LOGJAM_DHKEYSIZE_2048_BITS_ENABLED: true**.

You can set the parameter value to **true** or **false**.

Set **LOGJAM_DHKEYSIZE_2048_BITS_ENABLED** to **true** to avoid Logjam security vulnerability attack to disallow SHA-1 and Diffie-Hellman key exchange (DH) that is less than 2048 bits.

Following are the steps to change the value:

Changing the parameter value by using kubectl

1. Install the **kubectl** CLI.
2. Edit the **platform-auth-idp** ConfigMap.

   ```
   kubectl -n ibm-common-services edit configmap platform-auth-idp
   ```

3. Change the following attribute values as required:

   - Set **LOGJAM_DHKEYSIZE_2048_BITS_ENABLED** to **true** or **false**.

4. Save and close the ConfigMap.
5. Restart the **auth-idp** pods

   ```
   kubectl -n ibm-common-services delete pod -l k8s-app=auth-idp
   ```

6. Wait for some time. Then, check the status of the **auth-idp** pods. The status must show as **4/4 Running** for all the pods.

   ```
   kubectl -n ibm-common-services get pods | grep auth-idp
   ```

Changing the parameter values by using the console

1. Log in to the Red Hat® OpenShift® Container Platform console as a user with cluster administrator access.
2. From the navigation menu, click **Workloads** > **Config Maps.**
3. Search for **platform-auth-idp**.
4. Click **...** > **Edit Config Map**.
5. Change the **LOGJAM_DHKEYSIZE_2048_BITS_ENABLED** parameter value to **true** or **false**.
6. Click **Submit**.
7. From the navigation menu, click **Workloads > Deployments**.
8. Locate **auth-idp**.
9. Click **...** > **Edit Deployment**. A window for editing displays.
10. Click **Save** without making any change. This step is to reload the **auth-idp** pods with the latest ConfigMap values.
11. Click **auth-idp**.
12. Wait for some time. Then, check the status of the **auth-idp** pods in the **Pods** pane. The status of all the pods must show as **4/4** under the **Ready** field name.

Managing namespaces

Users are assigned to organizational units called namespaces.

Namespaces are also known as tenants or accounts. In IBM Cloud Pak® for AIOps, users are assigned to teams. You can assign multiple namespaces to a team. Users of a team are members of the team's namespaces.

A namespace in your product cluster corresponds to a single namespace in Kubernetes. All deployments, pods, and volumes that are created in a single namespace, belongs to the same Kubernetes namespace.

The following namespaces are reserved by IBM Cloud Pak for AIOps:

Table 1. IBM Cloud Pak for AIOps namespaces

Namespace	Description	Permission to access and deploy resources
cert-manager	Reserved for IBM Cloud Pak for AIOps certificate manager component.	Cluster administrator
default	Available when you install IBM Cloud Pak for AIOps and used as the default namespace for objects that do not specify a namespace. This namespace must not be used for any production workloads and must not be deleted.	Cluster administrator
icp-system	Reserved for IBM Cloud Pak for AIOps. This namespace must not be used for production workloads.	Cluster administrator
istio-system	Reserved for Istio platform services.	Cluster administrator
kube-public	Reserved by Kubernetes and IBM Cloud Pak for AIOps to store reference information that is available to any authenticated user. This namespace must not be used for production workloads.	Open access Only the cluster administrator can deploy resources
kube-system	Reserved for Kubernetes, your cluster, and other trusted workloads. This namespace must not be used for production workloads.	Cluster administrator
platform	Reserved for IBM Cloud Pak for AIOps. This namespace must not be used for production workloads.	Cluster administrator
services	Reserved for the IBM Cloud Automation Manager product.	Cluster administrator

The *Namespace overview* page in the consoledisplays the list of pod security policies that are associated to every namespace.

- Creating a namespace
- Removing a namespace
- Namespace limits

Creating a namespace

Applications must be created or deployed under a namespace in your product cluster. Namespaces are required to organize users and their applications.

Required user type or access level: Cluster administrator or team administrator

You can create a namespace by using the OpenShift Container Platform to create a Project. OpenShift Container Platform Projects are recognized as namespaces by IBM Cloud Pak® for AIOps. See Creating a Project in the OpenShift documentation for the procedure for creating a Project.

If the Kubernetes CLI (kubectl) is installed, you can use the **`kubectl create namespace`** command to create a namespace.

1. Provide a name for your namespace. The namespace name must meet the Kubernetes requirements for namespaces, including:

    - Be unique. You cannot use a duplicate namespace name.
    - Not exceed 63 characters in length.
    - Contain only lowercase letters, numbers, and the dash symbol (-).

2. Specify the pod security policy to be associated to your namespace. The pod security policy provides security aspects for pods that are created by applications that are installed on the namespace. The default applied pod security policy field displays the policy name that is applied across the cluster for all of the namespaces that are created.

   Attention: Multiple pod security policies can be associated with a namespace or ServiceAccount. Pods can resolve to any of the defined polices that are compatible, which might result in a less-restrictive policy. See Policy Order.

After the deployment completes, a new namespace or Project is displayed on the namespaces page.

Note: Be sure to scroll the **`All namespaces`** menu to view all of your namespaces.

Adding a label

You can use labels to create network policies that are based on a namespace selector. You cannot add labels for a namespace from the console. To add namespace labels, use the Kubernetes CLI.

1. Install the **`kubectl`** command line interface. See Installing the Kubernetes CLI (kubectl).
2. View a list of all namespaces.

   ```
   kubectl get namespaces
   ```

   The output resembles the following code:

   ```
   NAME           STATUS    AGE
   default        Active    6h
   dev            Active    2h
   kube-system    Active    6h
   qa             Active    2h
   ```

3. Add label.

   ```
   kubectl label namespaces dev team=dev
   ```

Removing a namespace

Remove namespaces that are no longer used.

Required user type or access level: Cluster administrator or team administrator

1. Log on to the console.
2. From the navigation menu, click **Identity and access** > **Teams and service IDs**.
3. Select the team name from the list of teams.
4. Select the **Resources** tab.
5. For the namespace that you want to remove, select **...** > **Remove**. A confirmation dialog box is displayed.
6. Click **Remove Namespace**.

Namespace limits

You can update the limits on your namespaces if you experience any constraints.

By default, the following limits are applied to IBM Cloud Pak® for AIOps namespaces:

Table 1. Namespace limits in IBM Cloud Pak for AIOps cluster

Namespace	CPU Maximum	CPU Minimum	CPU default requests	Memory Maximum	Memory Minimum	Memory default requests
cert-manager	800m	0m	300m	1Gi	0Mi	300Mi
default	500m	0m	200m	500Mi	0Mi	200Mi

Namespace	CPU Maximum	CPU Minimum	CPU default requests	Memory Maximum	Memory Minimum	Memory default requests
ibmcom	500m	0m	200m	500Mi	0Mi	200Mi
icp-system	500m	0m	200m	500Mi	0Mi	200Mi
kube-public	500m	0m	200m	500Mi	0Mi	200Mi
multicluster-endpoint	1500m	0m	300m	2Gi	0Gi	300Mi
services	500m	0m	200m	500Mi	0Mi	200Mi

- CPU represents how much processing power from your system that any Kubernetes pod can use. CPU is measured in cores. The values that are specified in the table are "millicores" or "millicpu". The maximum CPU of the default namespace is **`500 millicores`**.
- Memory is the amount of hard disk memory from your system that any Kubernetes pod can use. The constraints are based on the namespace where the Kubernetes pod runs.
- The maximum value caps the amount of CPU or memory that is used by the Kubernetes pods in a namespace. The minimum value forces every Kubernetes pod within the namespace to use at least the CPU and memory that are listed.

Updating limits

If a namespace requires more memory, for example, when you are deploying CAM in the **`services`** namespace, you can increase the limits by following the steps:

1. Edit the **`LimitRange`** object for the namespace you want to change the limits on. The following example shows the **`LimitRange`** object for the **`default`** namespace:

   ```
   kubectl edit LimitRange default-limit -n default
   ```

2. Change the maximum or minimum values of CPU, memory, or both:

   ```
   apiVersion: v1
   kind: LimitRange
   metadata:
      name: default-limit
   spec:
      limits:
      - max:
          cpu: "1000m"
          memory: "1Gi"
        min:
          cpu: "0m"
          memory: "0Mi"
        type: Container
   ```

3. Save your changes. Then, verify that the changes are applied by running the following command:

   ```
   kubectl describe namespace default
   ```

Disclaimers

- If a **`LimitRange`** does not set a default limit, the default limit implicitly becomes the maximum of the **`LimitRange`**. For example, the following LimitRange for the **`kube-public`** namespace has a maximum CPU of 100m and a maximum memory of 100Mi. The default for **`kube-public`** becomes CPU=100m and memory=100Mi.

   ```
   apiVersion: v1
   kind: LimitRange
   metadata:
     name: kube-public-limit
   spec:
     limits:
     - max:
         cpu: "100m"
         memory: "100Mi"
       min:
   ```

```
      cpu: "0m"
      memory: "0Mi"
    type: Container
```

- When a service is deployed to a namespace and it does not specify a resource limit on its container or containers, then the container inherits the default limits of the namespace.
- The implication of the previous two disclaimers is that if a container is deployed to a namespace without a resource limit, and the namespace does not have a default value set, then the limits of the container become the maximum limit of the namespace. As a result, you are not able to deploy the container because it shows that the namespace does not have enough memory, CPU, or both to support deploying it.

For more information about LimitRanges, see the Kubernetes documentation.

Managing Kubernetes Secret passwords with the IBM Cloud Pak CLI

You can enforce password requirements, change passwords, and restart the required pods and containers by using cloudctl for a set of Secrets.

Prerequisites

- **Required user type or access level:** Cluster administrator
- Install the IBM Cloud CLI (cloudctl).
- Install the Kubernetes CLI (kubectl).
- Log in to IBM Cloud Pak® for AIOps to generate tokens. The command prompts you for a password, account and namespace.

  ```
  cloudctl login -a https://<cluster-domain-name>:8443 -u <username> --skip-ssl-validation
  ```

Managing passwords

You can change several IBM Cloud Pak for AIOps passwords that are stored in Kubernetes Secrets. cloudctl has a command to change the password and restart services that use the password.

1. Choose from the following list of secrets which password you want to change:
   - platform-auth-idp-credentials: The cluster administrator access credentials
   - icp-mongodb-admin: The internal password for services that are using Mongo
   - platform-oidc-credentials: The internal password for authentication services
2. Run the command for the Secret that you want to change. You will receive a prompt for a password value and a prompt to confirm the change.

   Note: Do not use the special character `@` when you update the MongoDB password in **`icp-mongodb-admin`**.

   - platform-auth-idp-credentials

     ```
     cloudctl pm update-secret kube-system platform-auth-idp-credentials -d
     admin_password
     ```

   - icp-mongodb-admin

     ```
     cloudctl pm update-secret kube-system icp-mongodb-admin -d password
     ```

   - platform-oidc-credentials

     ```
     cloudctl pm update-secret kube-system platform-oidc-credentials -d WLP_CLIENT_SECRET
     ```

Optional: Managing password rules

Password rules are optional, regular expressions (regex) that are used to set and validate managed password values within a namespace. The **update-secret** command checks password rules before changing the values in the secret. It validates if rules exist on the namespace that the secret is in, and that the secret data element name change contains **pass** or **pwd** in any case combination.

Note: You can use the following regex expressions from the command line. Ensure that you wrap the expressions in single quotation marks:

Table 1. List of rules for Regex expressions

Rule	Regex
Min length 10	`^.{10,}`
Max length 10	`^.{0,10}$`
Length range	`^.{10,20}$`
Require lower	`.*[a-z].*`
Require upper	`.*[A-Z].*`
Require num	`.*[0-9].*`
Require spec	`.*[!@#$%^&*].*`

1. Set password rules with the following command:

   ```
   cloudctl pm password-rule-set <namespace> <rule_name> <rule_regex> <rule_desc>
   ```

   See the following example of a Regex expression:

   ```
   cloudctl pm password-rule-set default min_10 '^.{10,}' "minimum length of 10"
   OK
   ```

2. List the password rules.

   ```
   cloudctl pm password-rules <namespace>
   ```

   See the following example:

   ```
   cloudctl pm password-rules default

   Name   Description         Regex
   min10  minimum length 10  ^.{10,}
   OK
   ```

3. Remove the password rule.

   ```
   cloudctl pm password-rule-rm <namespace> <rule_name>
   ```

   See the following example:

   ```
   cloudctl pm password-rule-rm default min_10
   OK
   ```

Changing the Cloud Pak administrator access credentials

You can update the cluster administrator username and password.

Before you begin

- Install the IBM Cloud CLI (cloudctl).
- Install the OpenShift client.

- Install Platform UI (ibm-zen-operator) service.
- Ensure that you must have Cluster administrator access to update the cluster administrator username and password.

Changing the cluster administrator username

Complete the following steps to rename a default admin user in Platform UI:

1. Review the current admin and password.

   ```
   oc -n ibm-common-services get secret platform-auth-idp-credentials -o yaml | grep admin
   |xargs -l  bash -c 'echo -n "$0  " ; echo "$1" | base64 -d && echo'
   ```

   The sample output:

   ```
   admin_password:  IRXkjnsZg8nUUK8S2BBVFJeIOW0wVrhD
   admin_username:  admin
   ```

2. Run the **cloudctl** login command from any server that has access to your cluster where **oc** and **cloudctl** clients are installed.

   ```
   cloudctl login -a https://`oc get routes |grep -i cp-console | awk '{print $2}'` -u
   <username> -p <password> --skip-ssl-validation -n ibm-common-services
   ```

3. Use **cloudctl** password management (pm) command to change your username and to restart deployments. For example:

   ```
   cloudctl pm update-secret ibm-common-services platform-auth-idp-credentials -d
   admin_username=<username>
   ```

4. Update the **clusterrolebinding** role-based access control (RBAC) object with the new username.

   ```
   oc edit clusterrolebinding oidc-admin-binding
   ```

   Following is a sample **clusterrolebinding** RBAC object:

   ```
   apiVersion: rbac.authorization.k8s.io/v1
   kind: ClusterRoleBinding
   metadata:
     creationTimestamp: 2019-02-04T18:44:34Z
     name: oidc-admin-binding
     resourceVersion: "3162"
     selfLink: /apis/rbac.authorization.k8s.io/v1/clusterrolebindings/oidc-admin-binding
     uid: eab9c9c9-28ac-11e9-aca2-0050569a1e29
   roleRef:
     apiGroup: rbac.authorization.k8s.io
     kind: ClusterRole
     name: cluster-admin
   subjects:
   - apiGroup: rbac.authorization.k8s.io
     kind: User
     name: https://127.0.0.1:443/oidc/endpoint/OP#admin        <===========
   - apiGroup: rbac.authorization.k8s.io
     kind: User
     name: admin                                                <===========
   ```

   Note:

   - Replace the admin name in **https://127.0.0.1:443/oidc/endpoint/OP#admin** with the new name.
   - Change **OP#admin** to **OP#**.
   - Replace the admin name in **name: admin** with the new name.
   - Change **name: admin** to **name: <new admin user name>**. For example, updating to **cpadmin**.

   ```
   subjects:
   - apiGroup: rbac.authorization.k8s.io
     kind: User
     name: https://127.0.0.1:443/oidc/endpoint/OP#cpadmin        <===========
   - apiGroup: rbac.authorization.k8s.io
     kind: User
     name: cpadmin                                                <===========
   ```

5. Save the file.
6. Wait for few minutes for the **auth-idp** pod containers to start and validate the new user name.
    1. **cloudctl** login using the new user name. In the following code, **NewUsername** specifies the new user name and **password** specifies the admin password.

        ```
        cloudctl login -a https://`oc get routes -n ibm-common-services   |grep -i cp-console | awk '{print $2}'` -u <NewUsername> -p <password> --skip-ssl-validation -n ibm-common-services
        ```

    2. Test using the browser to login to the cp-console.

        ```
        oc get routes -n ibm-common-services |grep -i cp-console
        ```

Adding Platform UI user by using the Platform UI token

Administration panel with the **zen-cpp-operator** extension installed.

1. Get the Cloud Pak Platform route for accessing the common landing page by running the following command:

    ```
    oc get route -n ibm-common-services cpd -o jsonpath='{.spec.host}' && echo
    ```

    The response is your <https://<cluster_address>. <cluster_address> is the Cloud Pak Platform route.

    Sample output:

    ```
    cpd.apps.mycluster.mydomain.com
    ```

    Use this URL for **Zen-url** in the commnad of Step 3.

2. Run **cloudctl** tokens to get IAM access token.

    ```
    cloudctl tokens
    ```

    Note: When you use **cloudctl tokens**, you see the generated tokens. Keep a copy of the tokens. Do not run **cloudctl logout** command as it revokes the token.

3. Use the following command to get platform UI token (Zen_TOKEN) by using IAM token.

    ```
    curl -k X GET '<Zen-url>/v1/preauth/validateAuth' \
    -H "username: admin" \
    -H "iam-token: <iam-token>"
    ```

    It is valid only when Platform UI (ibm-zen-operator) service is installed. Use the token for **ZEN_TOKEN** in the command of Step 4.

4. Use the following command to add new user to Platform UI.

    ```
    curl --location --request POST 'https://ZEN_URL/usermgmt/v1/user'
    --header 'Authorization: Bearer ZEN_TOKEN'
    --header 'Content-Type: application/json'
    --data-raw '{
    "username": "newuser",
    "displayName": "UserFirstName UserLastName",
    "email": "user@in.ibm.com",
    "user_roles": [
    "zen_administrator_role",
    "zen_user_role"
    ]
    }
    ```

    Disable the default admin after you add new user to Platform UI. For more information, see Disabling the default admin user.

Changing the cluster administrator password

The cluster administrator password is stored in a Kubernetes Secret. cloudctl has a command to change the password and restart services that use the password.

To change your password run the following command:

```
cloudctl pm update-secret kube-system platform-auth-idp-credentials -d admin_password
```

The password must follow the defined password rules.

Password rules

Password rules are optional, regular expressions (regex) that are used to set and validate managed password values within a namespace. The **update-secret** command checks password rules before changing the values in the secret. It validates if rules exist on the namespace that the secret is in, and that the secret data element name change contains **pass** or **pwd** in any case combination.

Note: You can use the following regex expressions from the command line. Ensure that you wrap the expressions in single quotation marks:

Table 1. List of rules for Regex expressions

Rule	Regex
Min length 10	`^.{10,}`
Max length 10	`^.{0,10}$`
Length range	`^.{10,20}$`
Require lower	`.*[a-z].*`
Require upper	`.*[A-Z].*`
Require num	`.*[0-9].*`
Require spec	`.*[!@#$%^&*].*`

To list the password rules, run the following command:

```
cloudctl pm password-rules <namespace>
```

The rules are listed similar to the following example,

```
cloudctl pm password-rules default

Name   Description         Regex
min10  minimum length 10   ^.{10,}
OK
```

If you want to change the password rules, you can run the following command:

```
cloudctl pm password-rule-set <namespace> <rule_name> <rule_regex> <rule_desc>
```

For example,

```
cloudctl pm password-rule-set default min_10 '^.{10,}' "minimum length of 10"
```

If you need to remove the password rule, run the following command:

```
cloudctl pm password-rule-rm <namespace> <rule_name>
```

For example,

```
cloudctl pm password-rule-rm default min_10
OK
```

Certificate management

You can use IBM Cloud Pak for AIOps cert-manager to create and mount a certificate to a Kubernetes Deployment, StatefulSet, or DaemonSet. You can also create and add a certificate to a Kubernetes Ingress.

Issuer, *ClusterIssuer*, and *Certificate* are Kubernetes resource types that were introduced to support certificate generation and lifecycle management.

See the following list to learn how IBM Cloud Pak for AIOps cert-manager works:

- The Issuer signs new certificates and key pairs.
- The certificate object is similar to a certificate signing request.
- The actual X.509 certificate and key pair for TLS or authentication is stored as a Kubernetes Secret.
- The certificate is renewed automatically or can be renewed manually.

First, create an Issuer and then create a certificate that will be signed by that Issuer. IBM Cloud Pak for AIOps Certificate manager generates an X.509 certificate and key pair and stores it within a Kubernetes Secret.

Certificate manager supports IPv4 and IPv6 addresses.

For more information about Certificate manager and other configuration tools, see the following product documentation:

- Certificate management (operator)
- Accessing the foundational services endpoint certificate
- Replacing foundational services endpoint certificates
- Refreshing foundational services internal certificates
- Creating your own self-signed and CA Issuers
- Creating Certificate manager (cert-manager) certificates
- Customizing cert-manager certificates
- Viewing cert-manager resources
- Refreshing cert-manager certificates
- Using Vault to issue certificates
- Adding certificates by using the ECDSA algorithm for encryption
- Certificate Manager role-based access control (RBAC) support

Note: The apiVersion **`certmanager.k8s.io/v1alpha1`** in all cert-manager Custom Resources (Certificates, Issuers, and ClusterIssuers) is deprecated and replaced by **`cert-manager.io/v1`**.

Installing certificates

You can install certificates for IBM Cloud Pak for AIOps. The procedure for installing a certificate depends on the certificate or endpoint that you want to secure.

The three primary certificates covered here:

1. Red Hat OpenShift Container Platform certificate.
2. IBM Cloud Pak foundational services certificate.

   IBM Cloud Pak foundational services is a set of services that is installed with, and required by, IBM Cloud Pak for AIOps.

Red Hat OpenShift Container Platform console certificate

This certificate is for the Red Hat OpenShift Container Platform console and is typically only for the use of the cluster administrator, rather than other IBM Cloud Pak for AIOps users. It is critical that the certificate does not include the cp-console or the cpd-aiops routes in this certificate.

The default ingress certificate for Red Hat OpenShift Container Platform can have only a SAN of:

```
DNS.1 = *.apps.mycluster.example.com
```

IBM Cloud Pak foundational services

There are two options for installing a certificate. You can have a dedicated certificate for just the IBM Cloud Pak foundational services, or a combined certificate for it with IBM Cloud Pak for AIOps. Either option requires you to individually install the certificate: one to IBM Cloud Pak foundational services, and one to IBM Cloud Pak for AIOps.

The SAN in the combined case would be

```
DNS.1 = cp-console.apps.mycluster.example.com
DNS.2 = cpd-aiops.apps.mycluster.example.com
```

Note For the IBM Cloud Pak for AIOps cert, the certificate typically needs to be installed after you fully install IBM Cloud Pak for AIOps.

IBM Cloud Pak foundational services Certificate management

All certificates that are required by IBM Cloud Pak® foundational services are created during deployment by using certificate manager (cert-manager). A common CA issuer is created from a self-signed CA certificate (certificate authority) and leaf certificates are created by individual common services from the common CA issuer. Cert-manager automatically refreshes individual certificates. The CA certificate used by foundational services has a default duration of 2 years. Cert-manager will automatically refresh the CA certificate and the ibm-cert-manager-operator will automatically refresh the leaf certificates created from the CA certificate. You can customize the duration of the CA certificate based on the certification rotation schedule.

- foundational services cert-manager and community cert-manager version mapping
- Accessing the foundational services endpoint certificate
- Replacing the foundational services endpoint certificate
- Refreshing foundational services internal certificates
- IBM Certificate manager (cert-manager)
- Refreshing leaf certificates

Accessing the foundational services endpoint certificate

The foundational services endpoint certificate is stored within the Kubernetes Secret **ibmcloud-cluster-ca-cert** in the **ibm-common-services** namespace. The certificate can be imported into your client truststores to access the foundational services console and APIs. To retrieve and decode the certificate, run the following command:

```
oc get secret -n ibm-common-services ibmcloud-cluster-ca-cert -o jsonpath='{.data.ca\.crt}' |
base64 --decode
```

Your client truststore would need to be updated if the foundational services endpoint certificate is replaced or refreshed. Replacement or refresh is based on your certificate rotation schedule. For more information, see

- Replacing the foundational services endpoint certificate
- Refreshing foundational services internal certificates

Replacing the foundational services endpoint certificate

You can replace the certificate for the foundational services endpoint, which is the external endpoint that is used for access from outside the cluster to the console and API.

- Replacing the foundational services endpoint certificate (version 3.6.4 and later)
- Replacing the foundational services endpoint certificate (version 3.6.3 and earlier)

Replacing the foundational services endpoint certificate (version 3.6.4 and later)

Before you begin

Prepare and have your TLS certificate, TLS private key, and CA certificate ready. If needed, you can generate a TLS certificate by using OpenSSL. Your certificate must include the external hostname for the foundational services endpoint in the Subject

Alternative Name list. You can identify this hostname by using this command:

```
oc get route cp-console -n ibm-common-services -o jsonpath="{.spec.host}"
```

The following example configuration file and OpenSSL commands provide an example for how to generate a TLS certificate by using OpenSSL.

Example configuration file for generating a certificate

The following **csr.cnf** configuration file defines the configuration settings for generating certificates with OpenSSL.

```
# cat csr.cnf
[ req ]                # Main settings
default_bits = 2048       # Default key size in bits.
prompt = no               # Disables prompting for certificate values so the configuration
file values are used.
default_md = sha256       # Specifies the digest algorithm.
req_extensions = req_ext  # Specifies the configuration file section that includes any
extensions.
distinguished_name = dn   # Specifies the section that includes the distinguished name
information.

[ dn ]                # Distinguished name settings
C = US                    # Country
ST = New York             # State or province
L = Armonk                # Locality
O = IBM Cloud Private     # Organization
OU = IBM Cloud Pak        # Organizational unit
CN = cp-console           # Common name.

[ req_ext ]           # Extensions
subjectAltName = @alt_names # Subject alternative names

[ alt_names ]         # Subject alternative names
DNS.1 = <foundational service endpoint hostname>

[ v3_ext ]          # x509v3 extensions
authorityKeyIdentifier=keyid,issuer:always  # Specifies the public key that corresponds to the
private key that is used to sign a certificate.
basicConstraints=CA:FALSE                   # Indicates whether the certificate is a CA
certificate during the certificate chain verification process.
keyUsage=keyEncipherment,dataEncipherment   # Defines the purpose of the key that is contained
in the certificate.
extendedKeyUsage=serverAuth,clientAuth      # Defines the purposes for which the public key
can be used.
subjectAltName=@alt_names                   # Identifies the subject alternative names for the
identify that is bound to the public key by the CA.
```

OpenSSL commands for generating a certificate

The following OpenSSL commands are used with the preceding configuration file to generate the required TLS certificate.

1. Generate your certificate authority (CA) RSA private key:

   ```
   openssl genrsa -out ca.key 2048
   ```

2. Generate a self-signed CA certificate by using your CA key:

   ```
   openssl req -x509 -new -nodes -key ca.key -subj "/C=US/ST=New York/L=Armonk/O=IBM Cloud
   Private" -days 100 -out ca.crt
   ```

3. Generate the RSA private key for your certificate:

   ```
   openssl genrsa -out tls.key 2048
   ```

4. Generate the Certificate Signing request (CSR) by using the private key:

   ```
   openssl req -new -key tls.key -out tls.csr -config csr.cnf
   ```

5. Generate a signed certificate by using your CA certificate and key, and CSR:

```
openssl x509 -req -in tls.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out tls.crt -days
1000 -extensions v3_ext -extfile csr.cnf
```

6. Examine the certificate contents:

```
openssl x509  -noout -text -in ./tls.crt
```

Ensure that the common name on the certificate is the cluster CA domain for IBM Cloud Pak for AIOps.

Use your own certificate for the console and API

1. Update the **managementingress** CR so that **ibm-management-ingress-operator** does not re-create the **route-cert** certificate and the **route-tls-secret** secret.

```
oc -n ibm-common-services patch managementingress default --type merge --patch '{"spec":
{"ignoreRouteCert":true}}'
```

In the log of the **ibm-management-ingress-operator** pod, you see this message:

```
Not watching certificate: route-cert, IgnoreRouteCert is true.
```

2. Get your TLS certificate as described in Before you begin. Remember to include the external hostname for the foundational services endpoint in the **subjectAltName** list of your certificate.

3. Save the **cert**, **key**, and **ca-cert** files from your certificate in a directory. For example,

```
# ls -l
-rw-r--r-- 1 root root  2021 Oct 19 18:17 ca.crt
-rw-r--r-- 1 root root  1777 Oct 19 18:18 tls.crt
-rw-r--r-- 1 root root  1675 Oct 19 18:17 tls.key
```

You must have your full certificate chain if the CA certificate you bring is not self-signed. The full certificate chain includes all the CA certificates that signed each intermediate CA certificate that signed your CA certificate and your root CA. Save the full certificate chain in **ca.crt**.

4. Re-create the **route-tls-secret** with your certificate.

a. Delete the related certificate resource so that cert manager does not re-create your updated secret.

```
oc -n ibm-common-services delete certificates.v1alpha1.certmanager.k8s.io route-cert
```

b. Re-create the secret from your certificate.

```
oc -n ibm-common-services delete secret route-tls-secret
oc -n ibm-common-services create secret generic route-tls-secret --from-file=ca.crt=<your
path>/ca.crt  --from-file=tls.crt=<your path>/tls.crt  --from-file=tls.key=<your
path>/tls.key
```

c. Delete the **ibmcloud-cluster-ca-cert** secret. **ibm-management-ingress-operator** re-creates the **ibmcloud-cluster-ca-cert** secret and the **cp-console** route by using values from your new **route-tls-secret**.

```
oc -n ibm-common-services delete secret ibmcloud-cluster-ca-cert
```

5. Restart **auth-idp** pods.

```
oc -n ibm-common-services delete pod -l app=auth-idp
```

6. Restart **operand-deployment-lifecycle-manager** pods.

```
oc -n ibm-common-services delete pod -l name=operand-deployment-lifecycle-manager
```

7. Access the console to verify that the certificate is used.

Replacing the foundational services endpoint certificate (version 3.6.3 and earlier)

Before you begin

Prepare and have your TLS certificate, TLS private key, and CA certificate ready. If needed, you can generate a TLS certificate by using OpenSSL. Your certificate must include the external hostname for the foundational services endpoint in the Subject Alternative Name list. You can identify this hostname by using this command:

```
oc get route cp-console -n ibm-common-services -o jsonpath="{.spec.host}"
```

The following example configuration file and OpenSSL commands provide an example for how to generate a TLS certificate by using OpenSSL.

Example configuration file for generating a certificate (version 3.6.3 and earlier)

The following **csr.cnf** configuration file defines the configuration settings for generating certificates with OpenSSL.

```
# cat csr.cnf
[ req ]               # Main settings
default_bits = 2048       # Default key size in bits.
prompt = no               # Disables prompting for certificate values so the configuration
file values are used.
default_md = sha256       # Specifies the digest algorithm.
req_extensions = req_ext  # Specifies the configuration file section that includes any
extensions.
distinguished_name = dn   # Specifies the section that includes the distinguished name
information.

[ dn ]                # Distinguished name settings
C = US                    # Country
ST = New York             # State or province
L = Armonk                # Locality
O = IBM Cloud Private     # Organization
OU = IBM Cloud Pak        # Organizational unit
CN = cp-console           # Common name.

[ req_ext ]           # Extensions
subjectAltName = @alt_names # Subject alternative names

[ alt_names ]         # Subject alternative names
DNS.1 = <foundational service endpoint hostname>

[ v3_ext ]          # x509v3 extensions
authorityKeyIdentifier=keyid,issuer:always  # Specifies the public key that corresponds to the
private key that is used to sign a certificate.
basicConstraints=CA:FALSE                   # Indicates whether the certificate is a CA
certificate during the certificate chain verification process.
keyUsage=keyEncipherment,dataEncipherment   # Defines the purpose of the key that is contained
in the certificate.
extendedKeyUsage=serverAuth,clientAuth      # Defines the purposes for which the public key
can be used.
subjectAltName=@alt_names                   # Identifies the subject alternative names for the
identify that is bound to the public key by the CA.
```

OpenSSL commands for generating a certificate (version 3.6.3 and earlier)

The following OpenSSL commands are used with the preceding configuration file to generate the required TLS certificate.

```
1. Generate your certificate authority (CA) RSA private key:

 ```sh {: codeblock}
 openssl genrsa -out ca.key 2048
 ```
2. Generate a self-signed CA certificate by using your CA key:

 ```sh {: codeblock}
 openssl req -x509 -new -nodes -key ca.key -subj "/C=US/ST=New York/L=Armonk/O=IBM Cloud
Private" -days 100 -out ca.crt
 ```
3. Generate the RSA private key for your certificate:

 ```sh {: codeblock}
 openssl genrsa -out tls.key 2048
```
```

```
4. Generate the Certificate Signing request (CSR) by using the private key:

```sh {: codeblock}
openssl req -new -key tls.key -out tls.csr -config csr.cnf
```
5. Generate a signed certificate by using your CA certificate and key, and CSR:

```sh {: codeblock}
openssl x509 -req -in tls.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out tls.crt -days 1000 -extensions v3_ext -extfile csr.cnf
```
6. Examine the certificate contents:

```sh {: codeblock}
openssl x509  -noout -text -in ./tls.crt
```
Ensure that the foundational name on the certificate is the cluster CA domain for IBM Cloud Pak for AIOps.
```

Use your own certificate for the console and API (version 3.6.3 and earlier)

1. Update the **managementingress** CR so that **ibm-management-ingress-operator** does not re-create the **route-cert** certificate and the **route-tls-secret** secret. “Unmanaged” is case-sensitive.

   ```
   oc -n ibm-common-services patch managementingress default --type merge --patch '{"spec": {"managementState":"Unmanaged"}}'
   ```

   After this change, the **ibm-management-ingress-operator** pod will continue to run but it will do nothing. In the log of the pod, you see this message:

   ```
   do nothing for the managementingress: ibm-common-services/default because its state is unmanaged
   ```

2. Get your TLS certificate as described in Before you begin. Remember to include the external hostname for the foundational services endpoint in the **subjectAltName** list of your certificate.

3. Get the **destinationCACertificate** value from the current **cp-console** route.

   ```
   oc -n ibm-common-services get route cp-console -o jsonpath="
   {.spec.tls.destinationCACertificate}" > dest-ca.crt
   ```

4. Update the **cp-console** route.

   a. Save the **cert**, **key**, and **ca-cert** of your certificate in the same directory as the **destinationCACertificate**. For example,

   ```
   # ls -l
   total 68
   -rw-r--r-- 1 root root  2021 Oct 19 18:17 ca.crt
   -rw-r--r-- 1 root root  1168 Oct 19 18:28 dest-ca.crt
   -rw-r--r-- 1 root root  1777 Oct 19 18:18 tls.crt
   -rw-r--r-- 1 root root  1675 Oct 19 18:17 tls.key
   ```

   You must have your full certificate chain if the CA certificate you bring is not self-signed. The full certificate chain includes all the CA certificates that signed each intermediate CA certificate that signed your CA certificate and your root CA. Save the full certificate chain in **ca.crt**.

   b. Get the hostname from the current **cp-console** route. For example, **cp-console.apps.demo.cp.fyre.ibm.com**.

   ```
   oc -n ibm-common-services get route cp-console -o jsonpath="{.spec.host}"
   ```

   c. Regenerate the route spec. In the following command, replace **HOSTNAME** with the hostname you got in step b.

   ```
   oc -n ibm-common-services create route reencrypt cp-console --service=icp-management-ingress  --cert=./tls.crt  --key=./tls.key  --ca-cert=./ca.crt  --dest-ca-cert=./dest-ca.crt  --hostname=HOSTNAME  --insecure-policy='Redirect'  --dry-run=true -o yaml  > cp-console.yaml
   ```

d. Apply the change.

```
oc -n ibm-common-services  apply -f cp-console.yaml
```

5. Re-create the **route-tls-secret** with your certificate.

   a. Delete the related certificate resource so that cert manager does not re-create your updated secret.

   ```
   oc -n ibm-common-services delete certificates.v1alpha1.certmanager.k8s.io route-cert
   ```

   b. Re-create the secret from your certificate.

   ```
   oc -n ibm-common-services delete secret route-tls-secret
   oc -n ibm-common-services create secret generic route-tls-secret --from-file=ca.crt=<your
   path>/ca.crt  --from-file=tls.crt=<your path>/tls.crt  --from-file=tls.key=<your
   path>/tls.key
   ```

6. Re-create the **ibm-cloud-cluster-ca-cert** secret with your certificate.

   ```
   oc -n ibm-common-services delete secret ibmcloud-cluster-ca-cert
   oc -n ibm-common-services create secret generic ibmcloud-cluster-ca-cert --from-
   file=ca.crt=<your path>/ca.crt
   ```

7. Restart **auth-idp** pods.

   ```
   oc -n ibm-common-services delete pod -l app=auth-idp
   ```

8. Restart **operand-deployment-lifecycle-manager** pods.

   ```
   oc -n ibm-common-services delete pod -l name=operand-deployment-lifecycle-manager
   ```

9. Access the console to verify that the certificate is used.

10. You can use your own certificate and refresh the internal CA certificate as described in Refreshing foundational services internal CA certificate. Be sure to update the **cp-console** route after the internal CA certificate is refreshed. Refreshing foundational services internal CA certificate. updates the destination CA certificate. The destination CA certificate comes from the **icp-management-ingress-tls-secret** that is generated by the **management-ingress-cert** certificate.

    a. Get the updated **destinationCACertificate**.

    ```
    oc -n ibm-common-services get secret icp-management-ingress-tls-secret -o jsonpath=
    {.data."ca\.crt"} | base64 -d > dest-ca.crt
    ```

    b. Rerun step 4 to update the **cp-console** route by using the updated **dest-ca.crt**.

Refreshing IBM Cloud Pak foundational services internal certificates

- Refreshing foundational services internal certificates (IBM Cloud Pak foundational services Version 3.6.4 and later)
- Refreshing foundational services internal certificates (IBM Cloud Pak foundational services Version 3.6.3 and earlier)

Refreshing foundational services internal certificates (IBM Cloud Pak foundational services Version 3.6.4 and later)

A common CA issuer is created from a self-signed CA certificate (certificate authority) and leaf certificates are created by individual foundational services from the common CA issuer. Cert-manager automatically refreshes individual certificates. The CA certificate that is used by foundational services has a default duration of 2 years. Cert-manager will automatically refresh the CA certificate and the ibm-cert-manager-operator will automatically refresh the leaf certificates that are created from the CA certificate. You can customize the duration of the CA certificate based on the certification rotation schedule.

Changing the duration of the CA certificate and refreshing foundational services internal certificates

Complete the following steps to change the duration of the CA certificate and then refresh the foundational services internal certificates.

Note: `apiVersion: certmanager.k8s.io/v1alpha1` is deprecated. Use `apiVersion: cert-manager.io/v1`.

1. Update the `cs-ca-certificate` yaml file to add the `duration` and `renewBefore` parameters.

   ```
   duration: 17520h
   renewBefore: 240h
   ```

   In the following example, `duration` value is set to two years and the `renewBefore` value is set to 10 days.

   ```
   apiVersion: cert-manager.io/v1
   kind: Certificate
   metadata:
    name: cs-ca-certificate
    namespace: ibm-common-services
    labels:
     app.kubernetes.io/instance: ibm-cert-manager-operator
     app.kubernetes.io/managed-by: ibm-cert-manager-operator
     app.kubernetes.io/name: cert-manager
     certmanager.k8s.io/issuer-kind: Issuer
     certmanager.k8s.io/issuer-name: cs-ss-issuer
    spec:
     commonName: cs-ca-certificate
     isCA: true
     issuerRef:
      kind: Issuer
      name: cs-ss-issuer
     secretName: cs-ca-certificate-secret
     duration: 17520h
     renewBefore: 240h
    status:
     conditions:
     - lastTransitionTime: '2020-12-08T04:20:27Z'
     message: Certificate is up to date and has not expired
     reason: Ready
     status: 'True'
     type: Ready
     notAfter: '2021-03-08T04:20:27Z'
   ```

2. Delete the secret to refresh the CA certificate.

   ```
   oc -n ibm-common-services delete secret cs-ca-certificate-secret
   ```

   All leaf clusters that are created from cs-cs-certificate will be refreshed by the ibm-cert-manager-operator.

   Plan your certification rotation schedule by making note of the default CA certificate expiration, which is 2 years from deployment or the custom duration if set. The expiration date can be obtained with the following command.

   ```
   oc -n ibm-common-services get cert cs-ca-certificate
   ```

3. After refresh, if the monitoring console is not accessible then restart the alertmanager-ibm-monitoring-alertmanager-0, prometheus-ibm-monitoring-prometheus-0, and grafana pods.

   ```
   oc delete pod -l app=alertmanager -n ibm-common-services
   ```

   ```
   oc delete pod -l app=grafana -n ibm-common-services
   ```

   ```
   oc delete pod -l app=prometheus -n ibm-common-services
   ```

Refreshing foundational services internal certificates (IBM Cloud Pak foundational services Version 3.6.3 and earlier)

A common CA issuer is created from a self-signed CA certificate (certificate authority) and leaf certificates are created by individual foundational services from the common CA issuer. The CA certificate default duration is 90 days. You can customize this duration based on the certificate rotation schedule. Cert-manager will automatically refresh the CA certificate but the leaf certificates that are created from this CA certificate must be manually refreshed when the CA certificate is refreshed.

Complete the following steps to refresh the foundational services internal certificates.

1. Delete the secret to refresh the CA certificate.

   ```
   oc -n ibm-common-services delete secret cs-ca-certificate-secret
   ```

2. Refresh the leaf certificates based on the **cs-ca-certificate**.

   This step forces the leaf certificates to be updated with the new ca.

   ```
   mkdir secret_backup

   cd secret_backup

   oc get certs -o custom-columns=:spec.secretName,:spec.issuerRef.name --no-headers |egrep
   "cs-ca-clusterissuer|cs-ca-issuer" | while read secretName issuerName
   do
   oc get secret $secretName -o yaml -n ibm-common-services > secret.$secretName.yaml
   oc delete secret $secretName -n ibm-common-services
   done
   ```

   After the certificates are refreshed, all IBM Cloud Pak foundational services pods that mount these certificates will be automatically refreshed by cert-manager.

3. Restart the auth-idp, auth-pap, auth-pdp pods.

   ```
   oc delete pod -l app=auth-idp -n ibm-common-services

   oc delete pod -l app=auth-pap -n ibm-common-services

   oc delete pod -l app=auth-pdp -n ibm-common-services
   ```

4. Check and run the security-onboarding-job as-needed.

   a. Check the status of the security onboarding job.

   ```
   oc get jobs |grep security-onboarding
   ```

   If the job ran properly, it shows completion as 1/1. If a failure occurred, it shows 0/1.

   b. If a failure occurs, take a backup of the job and then delete the job to rerun it.

   ```
   oc get job <security-onboarding-job> -o yaml > security-onboarding-backup.yaml

   oc delete job security-onboarding
   ```

   c. Wait for the job to complete.

   ```
   oc get jobs

   # oc get jobs
   NAME                          COMPLETIONS   DURATION   AGE
   iam-onboarding                1/1           4m46s      157m
   oidc-client-registration      1/1           6m45s      3h11m
   security-onboarding           0/1           14s        14s
   [root@magesh-intel-inf ~]#

   # oc get jobs -w
   NAME                          COMPLETIONS   DURATION   AGE
   iam-onboarding                1/1           4m46s      158m
   oidc-client-registration      1/1           6m45s      3h11m
   security-onboarding           0/1           30s        30s
   security-onboarding           1/1           88s        88s
   ```

5. Refresh the **ibmcloud-cluster-ca-cert**.

   For foundational services versions before 3.6.2, the **ibmcloud-cluster-ca-cert** secret must be refreshed to pick up the refreshed **ca.crt**. The **ibm-management-ingress-operator** re-creates the secret.

   If you replaced the foundational services endpoint certificate by using procedure, Replacing the foundational services endpoint certificate, then do not perform this step.

Run the following command to delete the secret.

```
oc -n ibm-common-services delete secret ibmcloud-cluster-ca-cert
```

6. Re-create the **cp-console** route.

   For foundational services versions before 3.6.2, the **cp-console** route must be deleted and re-created to use the refreshed leaf certificate **route-tls** (secret **route-tls-secret**). The **ibm-management-ingress-operator** re-creates the route.

   Run the following command to delete the route.

   ```
   oc delete route cp-console -n ibm-common-services
   ```

7. (Optional) If you are using your own certificate and replaced the management-ingress certificate by using procedure, Replacing the foundational services endpoint certificate, then you must complete the following steps to update the **cp-console** route. The destination ca certificate is updated based on the refresh of the foundational services internal CA certificate.

   a. Obtain the updated destination CA certificate.

   ```
   oc get -o jsonpath={.data."ca.crt"} secret icp-management-ingress-tls-secret | base64 -d
   > dest-ca.crt
   ```

   b. Save the generated **cert**, **key**, and **ca-cert** of your certificate in the same directory.

   c. Regenerate the route spec.

   ```
   oc -n ibm-common-services create route reencrypt cp-console --service=icp-management-
   ingress     --cert=./tls.crt     --key=./tls.key     --ca-cert=./ca.crt     --dest-ca-
   cert=./dest-ca.crt     --hostname=cp-console.apps.demo.cp.fyre.ibm.com     --insecure-
   policy='Redirect'     --dry-run='client' -o yaml  > cp-console.yaml
   ```

   d. Apply the change.

   ```
   oc -n ibm-common-services  apply -f cp-console.yaml
   ```

8. After refresh, if the monitoring console is not accessible then restart the alertmanager-ibm-monitoring-alertmanager-0, prometheus-ibm-monitoring-prometheus-0, and grafana pods.

   ```
   oc delete pod -l app=alertmanager -n ibm-common-services
   ```

   ```
   oc delete pod -l app=grafana -n ibm-common-services
   ```

   ```
   oc delete pod -l app=prometheus -n ibm-common-services
   ```

Changing the duration of the CA certificate and refreshing foundational services internal certificates

Complete the following steps to change the duration of the CA certificate and then refresh the foundational services internal certificates.

1. Back up the certificate by running the following command:

   ```
   oc get certificate cs-ca-certificate -n ibm-common-services > cs-ca-
   certificate.yaml.bakup
   ```

2. Update the **cs-ca-certificate** yaml file and add the **duration** and **renewBefore** parameters.

   a. To update the **cs-ca-certificate** yaml file run the following command:

   ```
   oc edit certificate cs-ca-certificate -n ibm-common-services
   ```

   b. Add the **duration** and **renewBefore** parameters with the following values under the **spec** section in the **cs-ca-certificate** yaml file.

   ```
   duration: 17520h
   renewBefore: 240h
   ```

   In the following example, **duration** value is set to two years and the **renewBefore** value is set to 10 days.

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
name: cs-ca-certificate
namespace: ibm-common-services
labels:
app.kubernetes.io/instance: ibm-cert-manager-operator
app.kubernetes.io/managed-by: ibm-cert-manager-operator
app.kubernetes.io/name: cert-manager
certmanager.k8s.io/issuer-kind: Issuer
certmanager.k8s.io/issuer-name: cs-ss-issuer
spec:
commonName: cs-ca-certificate
isCA: true
issuerRef:
   kind: Issuer
   name: cs-ss-issuer
secretName: cs-ca-certificate-secret
duration: 17520h
renewBefore: 240h
status:
conditions:
- lastTransitionTime: '2020-12-08T04:20:27Z'
message: Certificate is up to date and has not expired
reason: Ready
status: 'True'
type: Ready
notAfter: '2021-03-08T04:20:27Z'
```

3. Save the **cs-ca-certificate** yaml file.

4. Delete the secret to refresh the CA certificate.

   ```
   oc -n ibm-common-services delete secret cs-ca-certificate-secret
   ```

5. Refresh the leaf certificates based on the **cs-ca-certificate**.

   Include this step if you refreshed **cs-ca-certificate** in the previous step, or if it was already refreshed by cert-manager. This step forces the leaf certificates to be updated with the new ca.

   ```
   mkdir secret_backup
   ```

   ```
   cd secret_backup
   ```

   ```
   oc get certs -o custom-columns=:spec.secretName,:spec.issuerRef.name --no-headers |egrep
   "cs-ca-clusterissuer|cs-ca-issuer" | while read secretName issuerName
   do
   oc get secret $secretName -o yaml -n ibm-common-services > secret.$secretName.yaml
   oc delete secret $secretName -n ibm-common-services
   done
   ```

   After the certificates are refreshed, all IBM Cloud Pak foundational services pods that mount these certificates will be automatically refreshed by cert-manager.

6. Restart the auth-idp, auth-pap, auth-pdp pods.

   ```
   oc delete pod -l app=auth-idp -n ibm-common-services
   ```

   ```
   oc delete pod -l app=auth-pap -n ibm-common-services
   ```

   ```
   oc delete pod -l app=auth-pdp -n ibm-common-services
   ```

7. Check and run the security-onboarding-job as-needed.

   a. Check the status of the security onboarding job.

   ```
   oc get jobs |grep security-onboarding
   ```

   If the job ran properly, it shows completion as 1/1. If a failure occurred, it shows 0/1.

   b. If a failure occurs, take a backup of the job and then delete the job to rerun it.

```
oc get job <security-onboarding-job> -o yaml > security-onboarding-backup.yaml

oc delete job security-onboarding
```

c. Wait for the job to complete.

```
oc get jobs

# oc get jobs
NAME                          COMPLETIONS   DURATION   AGE
iam-onboarding                1/1           4m46s      157m
oidc-client-registration      1/1           6m45s      3h11m
security-onboarding           0/1           14s        14s
[root@magesh-intel-inf ~]#

# oc get jobs -w
NAME                          COMPLETIONS   DURATION   AGE
iam-onboarding                1/1           4m46s      158m
oidc-client-registration      1/1           6m45s      3h11m
security-onboarding           0/1           30s        30s
security-onboarding           1/1           88s        88s
```

8. Refresh the **ibmcloud-cluster-ca-cert**.

   For foundational services versions before 3.6.2, the **ibmcloud-cluster-ca-cert** secret must be refreshed to pick up the refreshed **ca.crt**. The **ibm-management-ingress-operator** re-creates the secret.

   If you replaced the foundational services services endpoint certificate by using procedure, Replacing the foundational services endpoint certificate, then do not perform this step.

   Run the following command to delete the secret.

   ```
   oc -n ibm-common-services delete secret ibmcloud-cluster-ca-cert
   ```

9. Re-create the **cp-console** route.

For foundational services versions before 3.6.2, the **cp-console** route must be deleted and re-created to use the refreshed leaf certificate **route-tls** (secret **route-tls-secret**). The **ibm-management-ingress-operator** re-creates the route.

Run the following command to delete the route.

```
oc delete route cp-console -n ibm-common-services
```

10. (Optional) If you are using your own certificate and replaced the management-ingress certificate by using procedure, Replacing the foundational services endpoint certificate, then you must complete the following steps to update the **cp-console** route. The destination ca certificate is updated based on the refresh of the foundational services internal CA certificate.

a. Obtain the updated destination CA certificate.

```
  ```sh {: codeblock}
  oc get -o jsonpath={.data."ca.crt"} secret icp-management-ingress-tls-secret | base64 -d >
dest-ca.crt
  ```
```

b. Save the **cert**, **key**, and **ca-cert** of your certificate in the same directory. For example,

```
  ```sh {: codeblock}
  # ls -l
  total 68
  -rw-r--r-- 1 root root  2021 Oct 19 18:17 ca.crt
  -rw-r--r-- 1 root root  1168 Oct 19 18:28 dest-ca.crt
  -rw-r--r-- 1 root root  1777 Oct 19 18:18 tls.crt
  -rw-r--r-- 1 root root  1675 Oct 19 18:17 tls.key
  ```
```

c. Regenerate the route spec.

```
  ```sh {: codeblock}
   oc -n ibm-common-services create route reencrypt cp-console --service=icp-management-
ingress     --cert=./tls.crt     --key=./tls.key     --ca-cert=./ca.crt     --dest-ca-
```
```

```
cert=./dest-ca.crt     --hostname=cp-console.apps.demo.cp.fyre.ibm.com     --insecure-
policy='Redirect'     --dry-run='client' -o yaml  > cp-console.yaml
  ```
```

d. Apply the change.

```
  ```sh {: codeblock}
  oc -n ibm-common-services  apply -f cp-console.yaml
  ```
```

11. After refresh, if the monitoring console is not accessible then restart the alertmanager-ibm-monitoring-alertmanager-0, prometheus-ibm-monitoring-prometheus-0, and grafana pods.

```
oc delete pod -l app=alertmanager -n ibm-common-services
```

```
oc delete pod -l app=grafana -n ibm-common-services
```

```
oc delete pod -l app=prometheus -n ibm-common-services
```

For more information, see Customizing cert-manager certificates and Refreshing cert-manager certificates.

Creating your own self-signed and CA Issuers

To create certificates in IBM Cloud Pak® for AIOps that are managed by Certificate manager, you must first create an Issuer.

Create a self-signed Issuer, then use that Issuer to create a CA certificate. Your CA certificate can be managed by Certificate manager.

Note: **`apiVersion: certmanager.k8s.io/v1alpha1`** is deprecated. Use **`apiVersion: cert-manager.io/v1`**.

1. Create a self-signed Issuer. Use the following **`.yaml`** file to define a self-signed Issuer.

```
apiVersion: cert-manager.io/v1
kind: Issuer
metadata:
  name: hello-myself-tls
  namespace: foobar
spec:
  selfSigned: {}
```

2. After you create the self-signed Issuer, create a CA certificate that references the self-signed Issuer and specifies the **`isCA`** field.

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: hello-ca-tls
  namespace: foobar
spec:
  # name of the tls secret to store
  # the generated certificate/key pair
  secretName: hello-deployment-tls-ca-key-pair
  isCA: true
  issuerRef:
    # issuer created in step 1
    name: hello-myself-tls
    kind: Issuer
  commonName: "foo1.bar1"
  dnsNames:
  # one or more fully-qualified domain name
  # can be defined here
  - foo1.bar1
```

3. Edit the following sample of an Issuer that references the previous secret. Edit the **`name`** and **`namespace`** from the *metadata* section of the **`.yaml`** file. Be sure that **`secretName`** from the *spec* section matches the **`secretName`** from the previous step:
```

```
apiVersion: cert-manager.io/v1
kind: Issuer
metadata:
  name: hello-deployment-tls
  namespace: foobar
spec:
  ca:
    secretName: hello-deployment-tls-ca-key-pair
```

The **hello-deployment-tls-ca-key-pair** secret was created by Certificate manager and is managed, along with other certificates. This CA certificate can be used by the **hello-deployment-tls** Issuer.

Alternatively, you can also provide the CA certificate and private key instead of using Certificate Manager to create it. Run the following command to create your secret with your CA certificate and key pair files: **ca.crt** and **ca.key**.

```
kubectl create secret tls hello-deployment-tls-ca-key-pair --cert=ca.crt --key=ca.key --
namespace=foobar
```

Specify this Secret **hello-deployment-tls-ca-key-pair** in the Issuer **.yaml** file that you created earlier in step 3.

You can create a Certificate that is signed by this Issuer by changing the Certificate's **issuerRef.name** to the name of this Issuer **hello-deployment-tls**. For more information, see Adding a certificate to a Kubernetes workload.

See IBM Certificate manager (cert-manager) for more Certificate manager topics.

Creating Certificate manager (cert-manager) certificates

The Certificate manager service is used to issue and manage certificates for services. Certificate manager is based on the jetstack/cert-manager project.

- Adding a certificate to a Kubernetes workload
- Adding a certificate to a Kubernetes Ingress
- Customizing Certificate manager (cert-manager) certificates

Note: **apiVersion: certmanager.k8s.io/v1alpha1** is deprecated. Use **apiVersion: cert-manager.io/v1**.

Adding a certificate to a Kubernetes workload

1. Define an Issuer. You can define *Issuers* (namespace-scoped) or *ClusterIssuers* (cluster-scoped). Certificate manager service supports the following types of Issuers. For information about defining and creating Issuers, select the applicable links from the following list:

   - Self-signed and CA Issuers
   - Vault Issuer

2. To define the Certificate, edit the **metadata** where **<name>** is associated with the certificate and the **<namespace>** is where the certificate is created. Additionally, edit the **spec** section of the following example. The following example defines a certificate that uses the Self-signed Issuer from the Self-signed Issuer instructions in step one. Notice that the **<namespace>** of the certificate matches the **<namespace>** of the Issuer:

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: hello-deployment-tls-1
  namespace: foobar
spec:
  # name of the tls secret to store
  # the generated certificate/key pair
  secretName: hello-deployment-tls-1
  issuerRef:
    # Issuer Name
    name: hello-myself-tls
    # The default value is Issuer (i.e.
    # a locally namespaced Issuer)
    kind: Issuer
```

```
  commonName: "foo1.bar1"
  dnsNames:
   # one or more fully-qualified domain names
   # can be defined here
  - foo1.bar1
  ipAddresses:
   # one or more IP addresses can be defined here
  - 0.0.0.0
  - 127.0.0.1
```

Notes:

- If the **commonName** field is specified, then the **dnsNames** field is not mandatory and vice-versa.
- The value for **commonName** or the first value of **dnsNames** must be fewer than 64 characters long.
- The **secretName** must be unique to the namespace in which the certificate is created. In other words, it cannot have the same **secretName** as any other cert-manager certificate that exists in the same namespace.

3. Mount the Secret to the Deployment, DaemonSet, or StatefulSet.

   The Kubernetes Secret that contains the certificate is mounted to the file system in the same manner as any other secret. For more information, see the Kubernetes documentation.

Adding a certificate to Kubernetes Ingress

The Ingress Kubernetes resource type is used to expose services to an external network. Cert-manager-generated certificates can be added to Ingress resources. An NGINX Kubernetes Ingress point is provided for immediate use.

Handling Multiple Domain Names

Requests to multiple virtual hosts are handled by the same Ingress. Each virtual host can be terminated with its own certificates. In this case, the fully qualified domain name in the TLS/HTTPS request is used to identify the requested virtual host. The TLS-SNI protocol extension defines this process.

Complete the following procedure to secure the Kubernetes Ingress:

1. Define an Issuer. Certificate manager service supports the following types of Issuers. Click on the link to see how to define and create each:

   - Self-signed and CA Issuers
   - Vault Issuer

2. To define the Certificate, edit the **metadata** where **<name>** is associated with the Certificate and the **<namespace>** is where the Certificate is created. Additionally, edit the **spec** section of the following sample, which defines a Certificate that uses the CA Issuer from the CA Issuer instructions in step one. Notice the **<namespace>** of the Certificate matches the **<namespace>** of the Issuer:

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: hello-k8s-ingress-tls-1
  namespace: foobar
spec:
  # name of the tls secret to store
  # the generated certificate/key pair
  secretName: hello-k8s-ingress-tls-1
  issuerRef:
    # Issuer Name
    name: hello-ca-tls
    # The default value is Issuer (i.e.
    # a locally namespaced Issuer)
    kind: Issuer
  commonName: "foo1.bar1"
  dnsNames:
   # one or more fully-qualified domain names
   # can be defined here
  - foo1.bar1
```

   Cert-manager creates the certificate based on the certificate resource definition and stores it as a Kubernetes Secret.

3. Add the Secret to the Kubernetes Ingress. The following example defines a TLS-enabled Kubernetes Ingress that is integrated with cert-manager. Here, **`hello-k8s-ingress-tls-1`** matches the **`secretName`** that you previously defined and **`host`** matches the DNS name that you previously defined in the certificate.

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: hello-k8s-ingress-tls
  annotations:
    kubernetes.io/ingress.class: "nginx"
    ingress.kubernetes.io/rewrite-target: "/"
spec:
  tls:
  # k8s ingress defines different tls certificates
  # for each nginx server blocks.
  # k8s ingress default cert is used if
  # no host-specific secret specified
  - hosts:
  # this is the fully-qualified domain name
  # of the first server block
    - foo1.bar1
    # certificate hello-k8s-ingress-tls-1
    # is only used by foo1.bar1
    secretName: hello-k8s-ingress-tls-1
  rules:
    # each server block redirects request
    # to its own backend service
  - host: foo1.bar1
    http:
      paths:
      - backend:
          serviceName: hello-world-svc
          servicePort: 80
        path: /fb
```

Note: Certificates that are created by cert-manager are automatically renewed before expiration. For more information, see Refreshing Certificate manager (cert-manager) certificates.

For more Certificate manager topics, see IBM Certificate manager (cert-manager).

Customizing cert-manager certificates

Certificate manager service offers features that you can use to customize your certificates.

- Customizing certificate expiration
- Customizing certificate private key encoding
- Customizing certificate key usage

Note: **`apiVersion: certmanager.k8s.io/v1alpha1`** is deprecated. Use **`apiVersion: cert-manager.io/v1`**.

Customizing certificate expiration

The default expiration for all certificates is 90 days. The default time before expiration when the Certificate manager service renews the certificates is 30 days.

The Certificate manager service offers custom expiration times. The feature is offered by way of the new **`duration`** and **`renewBefore`** fields in a Certificate definition.

`duration` is how long the Certificate is valid. **`renewBefore`** is the time before the certificate expires when the Certificate manager service begins attempts to renew the certificate.

The following example of a Certificate definition has a validity period of 30 days (720 hours). Certificate manager service begins trying to renew this certificate 10 days (240 hours) before it expires.

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: hello-world-cert-1
  namespace: default
spec:
  secretName: hello-world-cert-1
  issuerRef:
    name: hello-ca-tls
    kind: Issuer
  commonName: "foo2.bar2"
  dnsNames:
  - foo2.bar2
  duration: 720h
  renewBefore: 240h
```

Read the following notes:

- The **duration** field must be no less than **1h**.
- The **duration** field must be greater than the **renewBefore** field.
- The parameters that are used for both the **duration** and **renewBefore** fields must be in the Golang ParseDuration format. Accepted time units include ["ns", "us" (or "µs"), "ms", "s", "m", "h"].
- If you specify a **duration** that is less than the default **renewBefore** time of 30 days (720 hours), then you must also specify the **renewBefore** field that is less than your **duration**.

Customizing certificate private key encoding

This version of cert-manager supports an extra format (PKCS#8) for private key encoding. If you want your certificates to encode their private keys in PKCS#8 format, specify the **keyEncoding** parameter in the **spec** field. For example:

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: example-pkcs8-certificate
spec:
  secretName: example-pkcs8-certificate-secret
  issuerRef:
      name: hello-ca-tls
      kind: Issuer
  commonName: "foo2.bar2"
  dnsNames:
  - foo2.bar2
  keyEncoding: pkcs8
```

Read the following notes:

1. The issuer does not have to have its key encoded in PKCS#8. In the previous example, the issuer that is created in section, Creating your own self-signed and CA issuers which has its key encoded in PKCS#1 format is used.
2. Remember that the resulting private key in the secret is generated in PKCS#8 format.
3. After you specify a certificate format, it can be changed to another format but you must refresh your certificate to have the change take effect. For more information, see Refreshing cert-manager certificates.
4. If the **keyEncoding** field is not specified, it defaults to PKCS#1 format

In addition, CA issuers can also have their private keys encoded in PKCS#8 format. Before only PKCS#1 format was supported. For more information, see Customizing issuers.

Following is a sample spec where the CA issuer is using a secret that contains a certificate and its private key. The private key is encoded in PKCS#8.

```
apiVersion: cert-manager.io/v1
kind: Issuer
metadata:
  name: pkcs8-ca-issuer
spec:
  ca:
    secretName: pk8-ca
```

Customizing certificate key usage

Certificates can now specify x509 actions are allowed/enabled for its key (key usage and extended key usage). Essentially it is one or more actions that certificates may be allowed to do. For more information about certificate key usage, see the RFC 5280 documentation.

Specify the actions that you want enabled by using the **usages** field. For example:

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: example-key-usage-certificate
  namespace: default
spec:
  secretName: example-key-usage-certificate
  issuerRef:
    name: hello-ca-tls
    kind: Issuer
  commonName: "foo2.bar2"
  dnsNames:
  - foo2.bar2
  usages:
  - digital signature
  - key encipherment
  - timestamping
```

Read the following notes:

1. The **usages** field is optional. If you do not specify a value in the **usages** field, the certificate includes default values, **key encipherment**, and **digital signature**.
2. However, if the field **isCA** is set to **true**, then the certificate includes **cert sign** as a default **usages** value.
3. Only the following values can be used in the **usages** field. Failure to issue the certificate occurs when you do not use the specified values.

```
  - signing
  - digital signature
  - content commitment
  - key encipherment
  - key agreement
  - data encipherment
  - cert sign
  - crl sign
  - encipher only
  - decipher only
  - any
  - server auth
  - client auth
  - code signing
  - email protection
  - s/mime
  - ipsec end system
  - ipsec tunnel
  - ipsec user
  - timestamping
  - ocsp signing
  - microsoft sgc
  - netscape sgc
```

Viewing cert-manager resources

IBM Cloud Pak for AIOps Certificate manager service creates five Kubernetes **CustomResourceDefinitions** (CRD):

1. Certificate
2. Issuer
3. ClusterIssuer

4. Orders
5. Challenges

You can use the **kubectl** command to view these Kubernetes resources.

- View the certificates in your cluster
- View the Issuers and ClusterIssuers in your cluster
- View the Orders and Challenges in your cluster

View the certificates in your cluster

To view all the certificates that are created and managed by IBM Cloud Pak for AIOps Certificate manager service, run the following command:

```
kubectl get certificate --all-namespaces
```

The command output provides the following information:

Table. Header descriptions

Header	Description
Namespace	The namespace where this certificate is located.
Name	The name of the certificate.
Ready	Whether the certificate is successfully issued and ready for use.
Secret	The name of the secret that contains the issued certificate, and private key for the certificate.
Age	When the certificate resource was first created in Kubernetes.
Expiration	When the certificate expires.

For more information about certificates, see kubectl flags.

For example, a **kubectl describe** gives you more information about the certificate:

```
kubectl describe certificate <certificate name> -n <certificate namespace>
```

View the Issuers and ClusterIssuers in your cluster

To view the Issuers or ClusterIssuers available in your cluster, run the following command:

```
# To view all Issuers
kubectl get issuer --all-namespaces

# To view ClusterIssuers
kubectl get clusterissuer
```

View the Orders and Challenges in your cluster

Orders and Challenges resources are created to issue ACME HTTP certificates, and are promptly deleted once the domain is verified and the certificate is issued. To view them, run the following command:

```
# To view all Orders
kubectl get orders --all-namespaces

# To view all Challenges
kubectl get challenges --all-namespaces
```

Refreshing cert-manager certificates

IBM Cloud Pak for AIOps Certificate manager service automatically refreshes certificates that are going to expire.

- Automatically refreshing leaf certificates created from a CS certificate
- Manually refreshing your certificates

- Disable restarting your service when a certificate is refreshed

Automatically refreshing leaf certificates created from a CA certificate (foundational services version 3.6.4 and later)

Cert-manager may be used to create a CA certificate. The CA certificate may then be used to create a CA Issuer. The CA issuer may then be used to create leaf certificates that are signed with the CA certificate. The cert-manager service will automatically renew certificates it creates and has been extended from the community source project to also support refresh of leaf certificates created from a CA certificate when then that CA certificate is refreshed. This is supported by the ibm-cert-manager-operator. A CA certificate and namespace may be specified and leaf certificates based on the CA certificate will be refreshed.

For more information, see Certificate Management Service settings

Manually refreshing your certificates

Certificates that are generated by IBM Cloud Pak for AIOps Certificate manager can be refreshed before IBM Cloud Pak for AIOps Certificate manager tries to refresh them by following these steps:

1. Determine the name of the Secret associated with your certificate. This information can be found by using the **`kubectl get certificate`** command and noting the namespace and Secret name next to your certificate.

   ```
   kubectl get certificate --all-namespaces
   ```

2. Delete the Secret associated with the certificate you want to refresh.

   ```
   kubectl delete secret <secret name> -n <namespace>
   ```

   NOTE: IBM Cloud Pak for AIOps Certificate manager re-creates the Secret for that certificate and automatically restarts any Pods associated with any Deployment, StatefulSet, and DaemonSet that uses that certificate.

   NOTE: When you refresh your certificate, you reset the **`duration`** and **`renewBefore`** values of your new certificate. Unless your new certificate specifies values for the **`duration`** and **`renewBefore`** parameters, the following defaults are applied:

   - **`duration`** value of **`90`** days
   - **`renewBefore`** value of **`30`** days
3. Wait for all the services that use that Secret to restart.

Disable restarting your service when a certificate is refreshed

When you refresh a certificate by using IBM Cloud Pak for AIOps Certificate manager service, it automatically restarts any Pods associated with any Deployment, StatefulSet, and DaemonSet that uses that Certificate.

You can disable this feature if you do not want IBM Cloud Pak for AIOps Certificate manager to restart the Pods associated with your Deployment, StatefulSet, or DaemonSet.

To disable the feature, provide the annotation **`certmanager.k8s.io/disable-auto-restart: "true"`** on your Kubernetes Deployment, StatefulSet, or DaemonSet yaml definition. For example,

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  annotations:
    certmanager.k8s.io/disable-auto-restart: "true"
...
```

DISCLAIMER: When IBM Cloud Pak for AIOps Certificate manager restarts your service, extra ReplicaSets are created. There can be old ReplicaSets that remain. To mitigate the problem of extra unused ReplicaSets, set the **`spec.revisionHistoryLimit`** in your Deployment to a reasonable number. If the **`spec.revisionHistoryLimit`** is not set, the default value is 10. For more information, see the Kubernetes documentation.

Using Vault to issue certificates

Certificate manager Issuer supports using the HashiCorp Vault server to create and issue certificates.

Vault is an external project. You must deploy and configure Vault to sign certificates before you follow this procedure. For more information about how to configure Vault as a certificate authority, see PKI Secrets Engine in HashiCorp Vault documentation.

Prerequisites

To use Vault Issuer, you must first set up a Vault server that is accessible to the IBM® Certificate manager in the OpenShift cluster. Certificate manager must be running in this cluster.

Configuring the certificate engine in Vault

The certificate engine in Vault must be configured in advance. Complete configuration and enablement steps.

Important: Certificate manager attempts to create certificates with a 90 day expiration when using the Vault Issuer, so the max_*ttl* needs to be at least 90 days (2160 hours). Do not use a max_*ttl* that is less than 30 days because Certificate manager attempts to renew certificates when the expiration is within 30 days.

Authenticating with the Vault server

To request Vault to sign certificates, the Issuer must be able to properly authenticate with Vault. Certificate manager provides multiple methods to authenticate with Vault.

Authenticating with an AppRole

You can use an **`AppRole`** to authenticate with Vault. An **`AppRole`** uses the Vault's internal role policy system. This authentication method requires the Issuer to have the **`secretID`**, which is a secret key, the **`RoleID`** of the role to assume, and the app role path. First, the **`SecretID`** key must be stored within a Kubernetes Secret that resides in the same namespace as the Issuer, or the Cluster Resource Namespace in case of a **`ClusterIssuer`**.

See the following YAML sample, which defines a Kubernetes Secret that contains your **`AppRole secretID`**:

```
apiVersion: v1
kind: Secret
type: Opaque
metadata:
  name: cert-manager-vault-approle
  namespace: <name of the namespace>
data:
  secretId: "ABCabc"
```

After you create the Secret, the Issuer is ready to be deployed with this Secret, and the key that stores the **`secretID`**.

See the following YAML sample, which deploys the Issuer with the secret reference and the key that stores the **`secretID`**:

```
apiVersion: cert-manager.io/v1
kind: Issuer
metadata:
  name: vault-issuer
  namespace: <name od the namespace>
spec:
  vault:
    path: pki_int/sign/example-dot-com
    server: https://vault.local
    caBundle: <base64 encoded caBundle PEM file>
    auth:
      appRole:
        path: approle
        roleId: "291b9d21-8ff5-..."
        secretRef:
```

```
          name: cert-manager-vault-approle
          key: secretId
```

Authenticating with Token

This method of authentication uses a token string that is granted from one of the authentication backends that Vault supports. The token expires, and you need to periodically refresh it. For more information, see Tokens in HashiCorp Vault documentation.

Note: Certificate manager does not automatically refresh the token. Make sure that you refresh the token regularly or consider automating this process.

First, the token needs to be stored inside a Kubernetes Secret that resides in the same namespace as the Issuer, or the Cluster Resource Namespace in case of a **`ClusterIssuer`**.

See the following YAML sample, which defines a Kubernetes Secret that contains your token:

```
apiVersion: v1
kind: Secret
type: Opaque
metadata:
  name: cert-manager-vault-token
  namespace: <name od the namespace>
data:
  token: "ABCabc"
```

After you create the Secret, the Issuer is ready to be deployed with this Secret, and the key that stores the token.

See the following YAML sample, which deploys the Issuer with the secret reference and the key that stores the token:

```
apiVersion: cert-manager.io/v1
kind: Issuer
metadata:
  name: vault-issuer
  namespace: <name od the namespace>
spec:
  vault:
    path: pki_int/sign/example-dot-com
    server: https://vault.local
    caBundle: <base64 encoded caBundle PEM file>
    auth:
      tokenSecretRef:
          name: cert-manager-vault-token
          key: token
```

Authenticating with Kubernetes Service Account token

You can configure Vault to authenticate applications by using the Kubernetes Service Account token.

For more information, see Managing Service Accounts in Kubernetes documentation. For more information about how to configure Vault to authenticate with a Service Account token, see Kubernetes Auth Method in the HashiCorp Vault documentation.

The Vault Issuer can use the Service Account token authentication only if Certificate manager has access to the token that is stored in the Kubernetes Secret. Kubernetes Service Account token is already stored in a Secret. You do not have to manually create it. However, you must make sure that this Secret is in the same namespace as the Issuer, or in the Cluster Resource Namespace for a **`ClusterIssuer`**.

This authentication method requires a **`role`**, which is the Vault role that the service account is to assume, and, optionally, a **`path`**, which is the authentication mount path that, by default, is set to **`kubernetes`**.

The following example shows the Service Account named **`my-service-account`** with the Secret reference and the key that stores the token. The key is set to **`token`** because the Secret was created by Kubernetes.

```
apiVersion: cert-manager.io/v1
kind: Issuer
metadata:
  name: vault-issuer
  namespace: <name of the namespace>
spec:
```

```
  vault:
    path: pki_int/sign/example-dot-com
    server: https://vault.local
    caBundle: <base64 encoded caBundle PEM file>
    auth:
      kubernetes:
        role: my-app-1
        mountPath: /v1/auth/kubernetes
        secretRef:
          name: my-service-account-token-hvwsb
          key: token
```

Using Vault as an Issuer to replace foundational services endpoint certificate

You can use Vault Issuer to manage the foundational services endpoint certificate. The external endpoint certificate is used to access the console and the APIs from outside the cluster.

In the following example, the in-cluster Vault server with Kubernetes Service Account authentication is used to manage the endpoint certificate.

Note: You can deploy and authenticate with Vault by using any preferred method.

Configuration of the certificate engine in Vault

In this example, the Vault server is deployed in the **vault** namespace. The service is created in the **vault** namespace with port **8200**. During configuration of the certificate engine in Vault, the CA certificate is put in the following path: **pki/sign/cs-ca-certificate**.

Authentication with Kubernetes service account

In this example, a Kubernetes authentication role is created to bind the PKI policy with a Kubernetes Service Account named **issuer** in the **ibm-common-services** namespace. **issuer-token-l42m9** is the token of the **issuer** Service Account.

Steps

The following steps are an example of how to set up Vault to issue foundational services endpoint certificates for the given configuration and

1. Generate the Vault Issuer to sign the endpoint certificate.

   ```
   apiVersion: cert-manager.io/v1
   kind: Issuer
   metadata:
     name: vault-issuer
     namespace: ibm-common-services
   spec:
     vault:
       auth:
         kubernetes:
           mountPath: /v1/auth/kubernetes
           role: issuer
           secretRef:
             key: token
             name: issuer-token-l42m9
       path: pki/sign/cs-ca-certificate
       server: 'http://vault.vault:8200'
   ```

2. Update the **managementingress** custom resource for the **ibm-management-ingress-operator** not to re-create the **route-cert** certificate and the **route-tls-secret** secret.

   ```
   oc -n ibm-common-services patch managementingress default --type merge --patch '{"spec":
   {"ignoreRouteCert":true}}'
   ```

   As the result, the following message is displayed in the **ibm-management-ingress-operator** logs:

   ```
   Not watching certificate: route-cert, IgnoreRouteCert is true.
   ```

3. Re-create the **route-tls-secret** with your certificate.

    1. Edit the related certificate resource so that Certificate manager can sign the secret by Vault Issuer.

        ```
        oc -n ibm-common-services edit certificates.v1alpha1.certmanager.k8s.io route-cert
        ```

        ```
        spec:
          ...
          duration: 8760h0m0s
          issuerRef:
            group: cert-manager.io
            kind: Issuer
            name: vault-issuer
          renewBefore: 24h0m0s
          secretName: route-tls-secret
          usages:
            - digital signature
            - key encipherment
            - server auth
        ```

    2. Delete the secret from the original certificate.

        ```
        oc -n ibm-common-services delete secret route-tls-secret
        ```

    3. Delete the **ibmcloud-cluster-ca-cert secret**. **ibm-management-ingress-operator** re-creates the **ibmcloud-cluster-ca-cert secret** and the **cp-console** route by using values from your new **route-tls-secret**.

        ```
        oc -n ibm-common-services delete secret ibmcloud-cluster-ca-cert
        ```

4. Restart the **auth-idp** pods.

    ```
    oc -n ibm-common-services delete pod -l app=auth-idp
    ```

5. Restart the **operand-deployment-lifecycle-manager** pod.

    ```
    oc delete pod -l name=operand-deployment-lifecycle-manager
    ```

6. Access the console and verify that the certificate is used.

Adding certificates by using the ECDSA algorithm for encryption

With certificate manager, you can encrypt with the ECDSA algorithm. A certificate can specify the key signing algorithm and the key size. Use the **keyAlgorithm** and **keySize** parameters to specify your customizations for the private key. If these parameters are not set, a 2048-bit RSA key is created.

- Creating a certificate with ECDSA
- Using ECDSA with Issuers

Note: **apiVersion: certmanager.k8s.io/v1alpha1** is deprecated. Use **apiVersion: cert-manager.io/v1**.

Creating a certificate with ECDSA

To create a certificate that uses the ECDSA algorithm for encryption, follow the procedure in Creating Certificate manager (cert-manager) certificates, but use the following sample where **keyAlgorithm** and **keySize** are required:

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: ecdsa-ca-cert
  namespace: default
spec:
  secretName: ecdsa-ca-secret
  keyAlgorithm: ecdsa
  keySize: 521
```

```
  isCA: true
  issuerRef:
    name: ss-issuer
    kind: Issuer
  commonName: foo1.bar1
  dnsNames:
  - foo1.bar1
```

Using ECDSA with Issuers

CA Issuer

1. See the following sample YAML, which uses Certificate manager to create a CA Issuer that uses ECDSA:

```
apiVersion: cert-manager.io/v1
kind: Issuer
metadata:
  name: ss-issuer
  namespace: default
spec:
  selfSigned: {}
```

2. Next, create a CA certificate that is issued from the self-signed Issuer and using the ECDSA key algorithm. See the following sample:

```
apiVersion: cert-manager.io/v1
kind: Certificate
metadata:
  name: ecdsa-ca-cert
  namespace: default
spec:
  secretName: ecdsa-ca-secret
  keyAlgorithm: ecdsa
  keySize: 521
  isCA: true
  issuerRef:
    name: ss-issuer
    kind: Issuer
  commonName: foo1.bar1
  dnsNames:
  - foo1.bar1
```

3. Edit the following sample to create the CA Issuer with the CA certificate:

```
apiVersion: cert-manager.io/v1
kind: Issuer
metadata:
  name: ecdsa-ca-issuer
  namespace: default
spec:
  ca:
    secretName: ecdsa-ca-secret
```

Vault Issuer

You can create certificates from a Vault Issuer that uses ECDSA keys instead of RSA by configuring the Vault PKI endpoint certificate update role. For more information about Vault Issuer, see Adding a certificate by using the Vault Issuer.

Go to your Vault server and run the following configuration command and edit the **key_bits** and **key_type**:

```
vault write auth/approle/role/my-role secret_id_ttl=8760h token_num_uses=0 token_ttl=20m
token_max_ttl=30m secret_id_num_uses=0 policies=pki_policy key_type=ec key_bits=521
```

When you create certificate resources, be sure to use a **keySize** that matches, or is greater than, the **key_bits** that you specify in your Vault server:

```
key_type=ec
key_bits=256, 384 or 521
```

Certificate Manager role-based access control (RBAC) support

RBAC support exists for Certificate Manager resources. For more information, see RBAC for Certificate Manager resources. All existing RBAC rules for other resources within IBM Cloud Pak for AIOps apply in addition to the Certificate Manager rules.

The ability to access Certificate Manager resources differs from the open source version of cert-manager.

Learn what each of the following roles can do with Certificate Manager:

- Cluster Administrator role
- Cloud Pak Administrator role
- Administrator role
- Operator role
- Other roles
- RBAC for Certificate Manager resources

Cluster Administrator role

Cluster Administrators within IBM Cloud Pak for AIOps can create, read/get/list, update, and delete all Certificate Manager resources (Certificates, Issuers, ClusterIssuers) within any namespace.

The implications include the ability to:

- Create Certificates that are signed by a ClusterIssuer
- Install or delete Helm charts that contain Certificate Manager resources

Cloud Pak Administrator role

Cloud Pak Administrators within your product can create, read/get/list, update, and delete only the Certificates and Issuers within the namespaces where Cloud Paks are deployed.

For the namespaces that the Cloud Paks are deployed in, the implications include:

- Cloud Pak Administrators cannot create Certificates that are signed by any ClusterIssuer
- The ability to deploy Helm charts:
  - Containing Certificates signed only by Issuers Containing Issuers
  - Not containing ClusterIssuers
  - Not containing Certificates signed by a ClusterIssuer
- The ability to delete Helm charts:
  - Containing Certificates that are signed by both Issuers and ClusterIssuers
  - Containing Issuers
  - Containing ClusterIssuers, but the ClusterIssuers are not removed with the rest of the resources in the Helm chart when the release is deleted

Administrator role

Administrators within IBM Cloud Pak for AIOps can create, read/get/list, update, and delete only the Certificates and Issuers within the namespaces they are Administrators in.

For the namespaces that the user is an **`Administrator`** in, the implications include:

- Administrators cannot create Certificates that are signed by any ClusterIssuer
- The ability to deploy Helm charts:
  - Containing Certificates signed only by Issuers
  - Containing Issuers
  - Not containing ClusterIssuers
  - Not containing Certificates signed by a ClusterIssuer
- The ability to delete Helm charts:
  - Containing Certificates that are signed by both Issuers and ClusterIssuers

- Containing Issuers
- Containing ClusterIssuers, but the ClusterIssuers are not removed with the rest of the resources in the Helm chart when the release is deleted

Operator role

Operators within IBM Cloud Pak for AIOps can create, read/get/list, and update only the Certificates and Issuers within the namespaces they are Operators in.

For the namespaces that the user is an `Operator` in, the implications include:

- Operators cannot create Certificates that are signed by any ClusterIssuer
- The ability to deploy Helm charts:
  - Containing Certificates signed only by Issuers
  - Containing Issuers
  - Not containing ClusterIssuers
  - Not containing Certificates signed by a ClusterIssuer

Other roles

All other roles, including but not limited to editors, viewers, and auditors, cannot create, read/get/list, update, or delete any Certificate Manager resources (Certificates, Issuers, and ClusterIssuers) within any namespace.

RBAC for Certificate Manager resources

IBM Cloud Pak for AIOps supports several roles. Your role determines the actions that you can perform.

Table 1. RBAC for Certificate Manager resources

Resource	Action	Administrator	Cloud Pak Administrator	Operator	Editor	Viewer
Certificates	Create	X	X	X		
Certificates	Read/Get/List	X	X	X		
Certificates	Update/Edit	X	X	X		
Certificates	Delete	X	X			
Issuers	Create	X	X	X		
Issuers	Read/Get/List	X	X	X		
Issuers	Update/Edit	X	X	X		
Issuers	Delete	X	X			
ClusterIssuers	Create, Read/Get/List, Update, Delete					

Secure Tunnel

A Secure Tunnel provides endpoint-to-endpoint connections across a hybrid network without opening firewall rules in an enterprise network. It allows access between different networks through TCP over HTTPS technology.

It can be applied in many use scenarios, such as MCMP, Runbooks, Infrastructure Automation, Slack Connection for AIOps, Instana, and Turbonomic.

The traffic through these connections is encrypted with HTTPS. A Secure Tunnel can control access to resources between different networks, with more granular control, collect all the operations, and traffic logins in audit records.

It is not necessary to modify any access rules and firewall configuration between existing infrastructure if you use a Secure Tunnel. It is like a traditional VPN, bridging two networks between the **Secure Tunnel server** and the **Secure Tunnel**

Connector.

To learn more about Secure Tunnel, see the following sections:

- Terminology
- Architecture
- Security
- Certificate expiration and renewal
- How to use Secure Tunnel

Terminology

To manage a connection, you need to understand the following concepts:

- **Secure Tunnel** a group of microservices that include the following services:
  - **UI Server**
    - Used to provide the static resources of the **Secure Tunnel console** and to handle some UI logic.
  - **API Server**
    - Verifies and processes the users operation request that comes from the **Secure Tunnel console**. The server operates the **Secure Tunnel Custom Resources(CRs)** according to the request.
  - **Controller**
    - Used to monitor updates to the **Secure Tunnel Custom Resources(CRs)** and to take actions according to the CR updates.
  - **Tunnel Connection**
    - The configuration data for bridging two networks.
  - **Application mapping**
    - The configuration data used to control:
      - which applications on the network of the **Secure Tunnel** side can be accessed by the **Secure Tunnel Connector** side.
      - which applications on the network of the **Secure Tunnel Connector** side can be accessed by the **Secure Tunnel** side.
    - The **Tunnel Connection worker** and **Secure Tunnel Connector** use this data to control the access to applications.
  - **Tunnel Connection worker**
    - The instance of the **Tunnel Connection**, one **Tunnel Connection** is created on one or more **Tunnel Connection worker** pods. It is also the server side of a TCP-over-HTTPS tunnel. Based on the OSS project.
  - **Secure Tunnel Connector**
    - The other side of a TCP over HTTP tunnel is installed on the peer network of the **Secure Tunnel** to bridge the network where the **Secure Tunnel** is installed. It is based on theOSS project.

Architecture

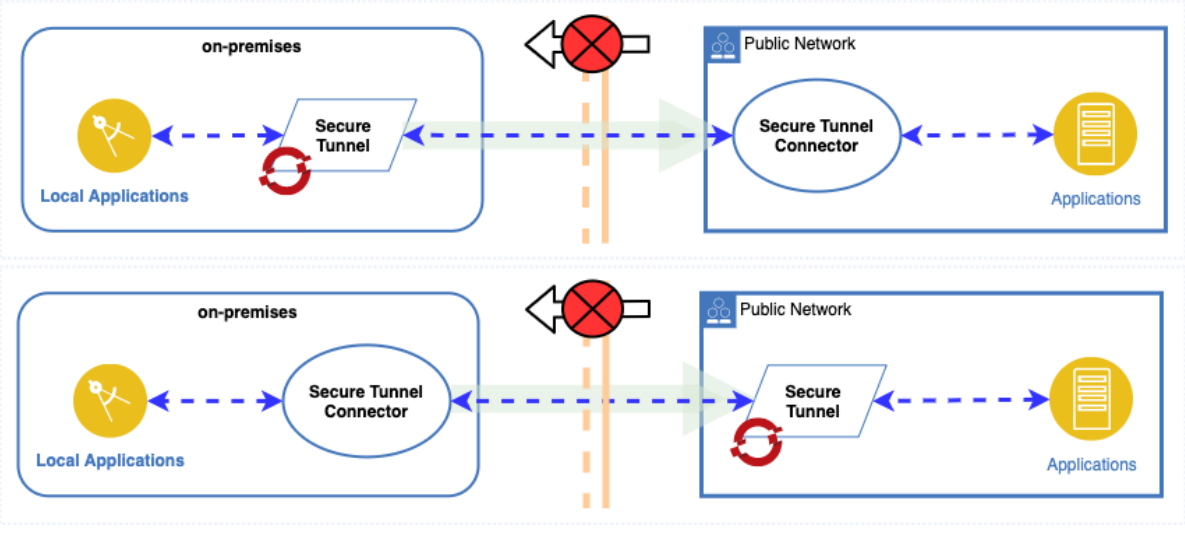


Figure. High-level design

Features

- Like a traditional VPN, Secure Tunnel can bridge two or more networks. Unlike VPN, it provides more fine-grained resource control.
- **Authentication and authorization** with the IBM Cloud Pak Automation console or Red Hat OpenShift Container Platform OAuth and RBAC system.
- Secure Tunnel provides two user interfaces
  - **Secure Tunnel console**: A customer can configure a **Secure Tunnel Connection** and control which applications can be accessed by the peer network from the console UI.
  - **Command line**: An automation script can configure a **Secure Tunnel Connection** and control which applications can be accessed by the peer network by using the `kubectl` or `oc` command line tools to operate the CRs of the **Secure Tunnel**.
- The connection is encrypted with Mutual Transport Layer Security(mTLS).
- Based on OSS project.

High-Level Design

- **Secure Tunnel** can be installed from the **Operator Hub** of the Red Hat OpenShift Container Platform console.
- **Secure Tunnel Connector** can be installed to an Red Hat OpenShift Container Platform console, Kubernetes (such as IBM Kubernetes Service), or host machine (VM or physical machine).
- It can bridge the network on the **Secure Tunnel** side and the network on the **Secure Tunnel Connector** side through a WebSocket connection between the **Secure Tunnel** and **Secure Tunnel Connector** with mTLS authentication.

You can use the **Secure Tunnel console** or **command line tools** to control:

- Which applications on the network of the **Secure Tunnel** side can be accessed by the **Secure Tunnel Connector** side.
- Which applications on the network of the **Secure Tunnel Connector** side can be accessed by the **Secure Tunnel** side.

Security

The following security measures make Secure Tunnel a safe tool to use:

- Authentication
  - IBM Cloud Pak for AIOps console accesses the Tunnel Connection worker with JWT authentication token.
  - The Connector is authenticated with Mutual Transport Layer Security (mTLS).
- Access control
  - Only IBM Cloud Pak for AIOps console users with the `Automation Administrator` or `Administrator` role can use the Secure Tunnel.
  - For more information about users and roles, see Managing user access control.
- Data encryption in communication
  - All traffic from out-cluster (Connector or console) is transported by HTTPS.
  - All application mapping traffic are encrypted by TCP over HTTPS.
- Auditing
  - All configuration changes (tunnel connection or application mapping) are stored in the log system for auditing. For more information, see Configuring tunnel audit logs.
  - All connections are stored in the log system for usage metrics and auditing.

Certificate expiration and renewal

If you renew the Secure Tunnel TLS certificate or it expires and the IBM Certificate Manager renews it, then the Secure Tunnel worker server automatically reloads the renewed certificate. However, the Secure Tunnel connector does not automatically

reload the renewed certificate. You must download a new Secure Tunnel connector installation script, and then uninstall and reinstall the Secure Tunnel connector. For more information, see the instructions in Install the Secure Tunnel Connector.

How to use Secure Tunnel

- To create a Secure Tunnel for an integration, see Creating Secure Tunnel Connections.
- To copy the application mapping URL that can be accessed by an integration, or view the configuration details and status of all tunnel connections and application mappings, see Viewing Secure Tunnel.
- To modify configurations for a Secure Tunnel, see Modifying Secure Tunnel.
- To audit Secure Tunnel operations and connections, see Auditing Secure Tunnel.

Installing Secure Tunnel

Use the following instructions to help you install Secure Tunnel for use with IBM Cloud Pak for AIOps.

Two methods are available to activate Secure Tunnel for use with IBM Cloud Pak for AIOps:

- Install Secure Tunnel from IBM Cloud Pak for AIOps
- Install Secure Tunnel without IBM Cloud Pak for AIOps (as a stand-alone component) to your Red Hat OpenShift cluster

Install Secure Tunnel from IBM Cloud Pak for AIOps

1. Ensure that IBM Cloud Pak for AIOps is installed.
2. An IBM Cloud Pak for AIOps console user with the **`Automation Administrator`** or **`Administrator role`** is needed to use Secure Tunnel. For more information about users and roles, see Managing user access control.
3. If you want to use Secure Tunnel to integrate with Slack or Microsoft Teams, you need to create a Slack connection or a Microsoft Teams connection on the IBM Cloud Pak for AIOps console. For more information, see Creating Slack connections and Microsoft Teams connections.
4. The Secure Tunnel component is not enabled by default at IBM Cloud Pak for AIOps installation time. Before you use Secure Tunnel, you need to enable the Secure Tunnel component.

   Two methods are available for enabling Secure Tunnel.

   - Console method:
     - Access Red Hat® OpenShift® Container Platform console.
     - Click **Operators** > **Installed Operators**.
     - Select **IBM Cloud Pak for AIOps** operator, and locate the **IBM Cloud Pak for AIOps** instance.
     - Set **Enable Secure Tunnel** to **`true`** to install Secure Tunnel.
   - YAML method:
     - Access Red Hat® OpenShift® Container Platform console.
     - From your Red Hat OpenShift console, click **Operators** > **Installed Operators**.
     - From the **Project** menu, select the project where you created the **IBM Cloud Pak for AIOps** instance.
     - Select **IBM Cloud Pak for AIOps** operator, and locate the **IBM Cloud Pak for AIOps** instance.
     - Click **Installation Item**. Switch to the **YAML** view, and update the YAML file by changing the enabled field of **`enableConnectionModule`** from **`false`** to **`true`**.

       ```
       spec:
         enableConnectionModule: true
       ```

       Note: If you can't find the connection and enabled fields in the YAML file, you need to add them to the YAML file manually.

- Ensure that all Secure Tunnel pods are in running state by running the following command:

  ```
  oc -n <namespace> get pod | grep tunnel
  ```

  Where **<namespace>** is the IBM Cloud Pak for AIOps namespace.

  The command result can resemble the following example:

  ```
  NAME                                                READY   STATUS    RESTARTS
  AGE
  ibm-secure-tunnel-operator-677c7b4b7-gqjbg          1/1     Running   0
  150m
  sre-tunnel-controller-7b9d59957b-rkxp7              1/1     Running   0
  147m
  sre-tunnel-tunnel-network-api-985df588c-z6m4z       1/1     Running   0
  147m
  sre-tunnel-tunnel-ui-mcmtunnelui-56c744598-qhlfw    1/1     Running   0
  147m
  ```

5. **(Optional) Enable EgressNetworkPolicy**: There is no egress firewall policy defined when you install Secure Tunnel, so outgoing traffic from workload pods to the internal and external network is unrestricted. If you require a more secure environment, complete the following steps:

   1. Go to your **OpenShift console** > **Operators** > **Installed Operators** > **IBM Secure Tunnel** > **Tunnel** > **sre-tunnel** > **YAML**.

   2. Add the following configuration to the YAML of the Tunnel CR to enable the egress NetworkPolicy:

      ```
      spec:
        enableEgressNetworkPolicy: true
        ...
      ```

      Note: This feature only supports Red Hat OpenShift Container Platform version 4.10 and above, and ensures there are not any restrictive EgressNetworkPolicies in place.

6. The Secure Tunnel Connector can be installed to both a cluster (OpenShift or IBM Kubernetes Service cluster) or a host machine (virtual machine or physical machine). Depending on which method you use, make the following preparations.

   - **If you want to install the Connector in a cluster (OpenShift or IBM Kubernetes Service cluster)**:

     - Ensure that the following minimum resource requirements are met:
       - 1 vCPU
       - 1 GB random access memory (RAM)
     - Ensure that you can access the cluster from a public network. The cluster domain name needs to be resolved by a public DNS.
     - Ensure that the cluster has a CA-signed TLS certificate so that the verification handshake of the Slack server or the Microsoft Teams server can be successful. A self-signed TLS certificate doesn't work for a Slack connection.
   - **If you want to install the Connector in a host machine (virtual machine or physical machine)**:

     - Ensure that the following minimum resource requirements are met:
       - 1 CPU
       - 1 GB random access memory (RAM)
     - Ensure that either Docker or Podman is installed, and running on the virtual machine or physical machine.
     - If the Connector is used to integrate with the Slack or MS Teams. You need to meet the following extra conditions:
       - Ensure that you can access the host machine from a public network. The hostname needs to be resolved by a public DNS.

       - Ensure that a CA-signed TLS certificate is available. This is used when you create the ChatOps connection. **Note**: If you have an IBM Cloud account, see Getting started with IBM Cloud Internet Services to apply for a **DNS name**, and see Preparing to order public certificates to apply for a CA-signed certificate. Extract the CA-signed certificate file, **`hostname.pem`**, for the certificate and **`hostname.key`** for the private key. If an intermediate file is in your CA-signed certificate file – such as hostname-intermediate.pem – concatenate this file with a cert file, such as hostname.pem, by running the following command:

```
cat hostname-intermediate.pem >> hostname.pem
```

Install Secure Tunnel without IBM Cloud Pak for AIOps (as a stand-alone component) to your OpenShift cluster

1. Ensure that your cluster meets the following requirements:

   - Supported Red Hat OpenShift Container Platform versions
   - Hardware requirements. Ensure that the following minimum resource requirements are met:
     - 2 vCPU
     - 2 GB random access memory (RAM)
2. Create the catalog source with the OpenShift console:

   1. Log in to your Red Hat OpenShift cluster's console.

   2. Add the IBM Operators CatalogSource. Click the **plus** icon in the upper right corner to open the *Import YAML* dialog box. Paste in the following content, and then click **Create**:

      ```
      apiVersion: operators.coreos.com/v1alpha1
      kind: CatalogSource
      metadata:
        name: ibm-operator-catalog
        namespace: openshift-marketplace
      spec:
        displayName: ibm-operator-catalog
        publisher: IBM Content
        sourceType: grpc
        image: icr.io/cpopen/ibm-operator-catalog:latest
      ```

   3. Go to **Administration** > **Cluster Settings**. Under **Global Configuration** > **OperatorHub** > **Sources**, verify that the `ibm-operator-catalog` CatalogSource object is present.

      Note: For more information, see Catalog Sources

3. Install the Secure Tunnel operator with the Red Hat OpenShift console:

   1. Log in to your Red Hat OpenShift cluster console.
   2. Click **Operators** > **OperatorHub**. The *OperatorHub* page is displayed.
   3. In the **All Items** field, enter "IBM Secure Tunnel". The Secure Tunnel operator is displayed.
   4. Click the **IBM Secure Tunnel** tile. The *IBM Secure Tunnel* window is displayed.
   5. Click **Install**. The *Install Operator* page is displayed.
   6. Enter the following values:
      1. Set the **Installation mode** to **All namespaces on the cluster(default)**.
      2. Set **Update Channel** to **v4.2**.
      3. Set **Approval Strategy** to **Automatic**.
   7. Click **Install** and wait for the Secure Tunnel operator to install.
   8. Verify that the IBM Cloud Pak for AIOps operator is successfully installed.
      1. Navigate to **Operators** > **Installed Operators**, and select your project from the *Projects* dropdown. Secure Tunnel operator in the project is listed with a status of `Succeeded`.
4. Install Secure Tunnel

   1. To create a Secure Tunnel project (namespace) with the Red Hat OpenShift console click **Home** > **Projects**. Select **Create Project**, specify the name of the project that you want to create (for example `tunnel-stand-alone`) and click **Create**.

      Note: For more information, see Namespace.

   2. Create the entitlement key secret with the Red Hat OpenShift console. Complete the following steps to create a docker-registry secret to enable your deployment to pull the Secure Tunnel images from the IBM® Entitled Registry:

      1. Obtain the entitlement key that is assigned to your IBMid. Log in to MyIBM Container Software Library with the IBMid and password details that are associated with the entitled software.
      2. In the Entitlement key section, select **Copy key** to copy the entitlement key to the clipboard.

3. From your Red Hat OpenShift console, click **Workloads** > **Secrets**.
4. From the **Project menu**, select the **project** that you created earlier in **Create a Secure Tunnel project (namespace)**.
5. Click the **Create** button, and select Image Pull Secret from the menu. The Create Image Pull Secret form is displayed. Enter the following values and then click Create.
    - **Secret Name:** `secure-tunnel-pull-secret`
    - **Authentication Type:** Image Registry Credentials
    - **Registry Server Address:** `cp.icr.io`
    - **Username:** `cp`
    - **Password:** Use the entitlement key that you copied earlier.

      Note: For more information, see Entitlement Keys

3. Create an instance of the Secure Tunnel with the Red Hat OpenShift console. Secure Tunnel is supported by namespace isolation, so you can create multiple Secure Tunnel instances in different namespaces, for different users. Secure Tunnel also uses the Red Hat OpenShift OAuth&RBAC for authentication and authorization. You can configure the permissions for each Secure Tunnel instance user.

    1. From your Red Hat OpenShift console, click **Operators** > **Installed Operators**.
    2. From the **Project** menu, select the project that you want to create the Secure Tunnel instance in. Use the project that you created earlier in **Create a Secure Tunnel project (namespace)**.
    3. Select **IBM Secure Tunnel** operator.
    4. Click **Create Installation** on the **Tunnel** box. The default **Form View** is displayed. Enter the following values:

        - **Name:** Specify the name for Secure Tunnel instance.
        - **License:** Expand the *License* section and read the agreement. Toggle the *License Acceptance* switch to *True* to accept the license.
        - **Image Pull Secret:** Select the `secure-tunnel-pull-secret` secret that you created in the step **Create the entitlement key secret**.
        - **runAsStandAlone**: If the **IBM Certificate manager** is not installed in your environment, toggle this switch to *True* and it will create the TLS Certificate that is used by the Secure Tunnel without the **IBM Certificate manager**.
        - **Size:** Select the size that you require for your Secure Tunnel installation.
        - **Audit Receiver:** Refer to the audit log guide
        - (Optional) **Enable EgressNetworkPolicy:** There is no egress firewall policy defined when you install Secure Tunnel, so outgoing traffic from workload pods to the internal and external network is unrestricted. If you require a more secure environment, toggle this switch to `True` to enable the egress NetworkPolicy. ***Note:** This feature only supports Red Hat OpenShift Container Platform version 4.10 and above, and ensures that there are no restrictive EgressNetworkPolicies in place.

    Refer below to the tunnel parameters with their description, variable type and default values.

    Tunnel settings:

Tunnel Settings

Parameter	Description	Default value	Type
size	The sizing of the Secure Tunnel. Supported values: "small", "medium", "large".small: replica is 1 for all operands, no HA;medium: replica is 3 with HA enabled;large: replica is 3 with HA enabled and with extra capacity for CPU/memory.	small	String

Parameter	Description	Default value	Type
license.accept	Expand the License section and read the agreement. Toggle the License Acceptance switch to True to accept the license.	false	Boolean
uiReplicas	The replicas of the UI server pods.	Depends on size	int
apiReplicas	The replicas of the API server pods.	Depends on size	int
controllerReplicas	The replicas of the controller server pods.	Depends on size	int
workerReplicas	The replicas of the worker server pods per tunnel connection.	Depends on size	int
pullSecret	The pull secret of the Secure Tunnel docker images.	secure-tunnel-pull-secret	String
runAsStandAlone	If you want to install the Secure tunnel without AIOps, toggle this to True	false	Boolean
showInContainerLog	Show audit logs in the container log.	true	Boolean
auditReceiver.mongoDB.mongoAddress.name	The name of the audit log MongoDB receiver.	null	String
auditReceiver.mongoDB.mongoAddress.mongoAddress	The audit log MongoDB receiver address. Example: `<IP>:27017`	null	String
auditReceiver.mongoDB.mongoAddress.passwordSecretName	The Kubernetes secret that contains the audit log MongoDB receiver login username and password. Create it in the Secure Tunnel namespace. It should contain "user" and "password" data items.	null	String
auditReceiver.mongoDB.mongoAddress.mongoDatabaseName	The database name of the audit log MongoDB receiver	null	String
auditReceiver.mongoDB.mongoAddress.tlsSecret The tls Kubernetes secret name of the audit log MongoDB receiver. Create it in the Secure Tunnel namespace. It should contain "ca.crt", "tls.crt" and "tls.key".	null	String	
auditReceiver.syslog.name	The name of the audit log Syslog receiver.	null	String
auditReceiver.syslog.syslogAddress	The server address of the audit log Syslog receiver.	null	String
defaultTunnelWorkerResources.limits.cpu	The default Tunnel Worker cpu limit.	3000m	String
defaultTunnelWorkerResources.limits.memory	The default Tunnel Worker memory limit.	3072Mi	String
defaultTunnelWorkerResources.requests.cpu	The default Tunnel Worker cpu requests.	100m	String
defaultTunnelWorkerResources.requests.memory	The default Tunnel Worker memory requests.	128Mi String	
applicationMappingPortRang	Application Mapping listen port range. For example: 30000-65000	30000-65000	String

Parameter	Description	Default value	Type
enableEgressNetworkPolicy	If you require a more secure Secure Tunnel environment you can enable this item. Then it will create the egress network policy automatically. Note: This only works on OCP 4.10 or later.	false	Boolean
connectionTLSCertDurationTime	By default, the TLS certificate that is used by the Tunnel connection. Duration time is 87600h0m0s(10 years). Update this item if you want to change the duration time.	87600h0m0s(10 years)	String

5. Click **Create**. This creates the Secure Tunnel instance.
6. After a few minutes, select the project that you specified for your Secure Tunnel instance from the **Project** menu. Navigate to **Operators** > **Installed Operators** > **IBM Secure Tunnel** and then click the **Tunnel** tab. Under **Installations**, look for the entry with the name that you specified for your Secure Tunnel instance, and verify that it has a *Status* of **Phase: Running**.
7. Refer to: Production installation **Note:** One project(namespace) can only create one Secure Tunnel instance.

5. Access the Secure Tunnel console:
   1. From your Red Hat OpenShift console, click **Networking** > **Routes**.
   2. From the **Project** menu, select the project where you created the **Secure Tunnel** instance.
   3. Select the `sre-tunnel-tunnel-ui`.
   4. Click the link of the **Location**. The Secure Tunnel console opens. Log in with your Red Hat OpenShift Username and password.

Uninstalling Secure Tunnel

- **If the Secure Tunnel was installed from IBM Cloud Pak for AIOps**, complete the following steps to disable it:
  1. From your Red Hat OpenShift console, click **Operators > Installed Operators**.
  2. From the **Project** menu, select the project where you created the **IBM Cloud Pak for AIOps** instance.
  3. Select the **IBM Cloud Pak for AIOps** operator, and locate the **IBM Cloud Pak for AIOps** instance.
  4. Click **Installation Item**. Switch to the **YAML** view, and update the YAML file by changing the enableConnectionModule field of connection from `true` to `false`.

     ```
     spec:
       enableConnectionModule: false
     ```

- If the Secure Tunnel was installed from the Operator hub as stand-alone, complete the following steps to uninstall it:
  1. From your Red Hat OpenShift console, click **Operators > Installed Operators**.
  2. From the **Project** menu, select the project where you installed the **IBM Secure Tunnel**.
  3. Select **IBM Secure Tunnel** operator, and locate the Tunnel instance.
  4. Delete the `sre-tunnel` (by default) instance of the Tunnels.
  5. Uninstall the IBM Secure Tunnel operator from **Operators > Installed Operators**

Note: There can be some resources, such as the CRDs and TLS cert secret of the secure tunnel that will not be deleted. If you want to completely remove them, you can download the `uninstall-secure-tunnel.sh` and run the command `./uninstall-secure-tunnel.sh -n <the namespace of the secure tunnel>` to do that.

Install the Secure Tunnel Connector

Install the Secure Tunnel Connector for the network being connected to a cluster, Red Hat® OpenShift® on IBM Cloud® service (ROKS) cluster, or a host machine (VM or physical). With the Secure Tunnel Connector installed, you can establish a secure tunnel between the network of the cluster where you installed the **Secure Tunnel** (server), such as your IBM Cloud Pak for AIOps cluster, and the cluster where you installed the **Secure Tunnel Connector**.

Important Depending on how you plan to create your Secure Tunnel connection, the Secure Tunnel Connector can be installed automatically. For instance, if you are connecting IBM Cloud Pak for AIOps to a Red Hat OpenShift Container Platform cluster for a ChatOps application, the Secure Tunnel Connector can be installed automatically. For more information, see Creating a Secure Tunnel connection.

Note: If your Secure Tunnel server is installed on an OpenShift cluster with a domain name that is not resolvable from DNS, then Secure Tunnel might not be able to access it through the OpenShift URL. You might find that the Tunnel Connector fails to connect to the Secure Tunnel server. You need to log in to the host machine, on which Tunnel Connector is installed, and add the IP and domain (for example, 9.30.111.xxx console-openshift-console.apps.test.cp4aiops.com) to **`/etc/hosts`** to resolve all the domain names. Follow these steps:

1. Log in to the host machine on which Tunnel Connector is installed.
2. Find out all the domain names to be added, run the following command in the **`tunnel-connector-install-scripts`** directory:

   ```
   cat install-connector.sh | grep "-e SERVER=" | head -n 1
   ```

   You find outputs like this:

   ```
   SERVER='https://r8b74ff6d23d84fd50-cp4aiops.apps.svt-airgap-1.svt.cp4aiops.com/sre-tunnel-tunnel/ws;https://r8b74ff6d23d84fd51-cp4aiops.apps.svt-airgap-1.svt.cp4aiops.com/sre-tunnel-tunnel/ws;https://r8b74ff6d23d84fd52-cp4aiops.apps.svt-airgap-1.svt.cp4aiops.com/sre-tunnel-tunnel/ws;' \,
   ```

3. Add each domain in the output to the **`/etc/hosts`** with the IP address of your OpenShift cluster(for example **`9.30.111.xxx r8b74ff6d23d84fd50-cp4aiops.apps.svt-airgap-1.svt.cp4aiops.com`**)
4. If you already installed Tunnel Connector before doing these steps, then you need to restart Tunnel Connector by using the following command:

   ```
   docker/podman restart <the container id>
   ```

5. If you completed the previous steps and the tunnel connection console UI displays the **Ready** status, but your application, which is using the Secure tunnel, still does not work, check whether you added all domains to **`/etc/hosts`**.
6. If you selected **`Cloud`** when you created the Tunnel Connector, you need to re-create the Tunnel Connector, select **`Host machine`**, download the Tunnel Connector installation package, and install this package to a host machine.

Prerequisites

Download the Secure Tunnel Connector installation scripts:

- From the console, go to **Manage connections**, locate the tunnel connection name, and click **Action** > **Install Connector**.
- Obtain the installation package **`tunnel-connector-install-scripts.tar.gz`** by clicking **Download**. Or you can click **Copy**, and then run the copied commands on a Linux system.

  Note: The installation script is for only a specific **Secure Tunnel Connection**, and cannot be used for other Secure Tunnel Connections.

Installing the Secure Tunnel connector to a cluster

1. Transfer the installation package **`tunnel-connector-install-scripts.tar.gz`** to a virtual machine where you can log in and access the cluster where you want to install the Connector.

   Note: Ensure that the OpenShift **`oc`** or **`kubectl`** tool is installed in the virtual machine.

2. Log in to the cluster on the virtual machine with the **`oc login`** command, and unpack the installation package.

3. Create the project (namespace) where you want to install the Connector.
4. Unpack the **tunnel-connector-install-scripts.tar.gz** package.
5. Change to the **tunnel-connector-install-scripts** directory.
6. Run the following command to install the Connector:

   ```
   ./install-connector.sh --namespace <namespace> --accept-license true
   ```

7. Verify whether the Connector is installed by running the following command:

   ```
   oc get po -n <namespace>
   ```

Installing the Secure Tunnel connector to a host machine (virtual machine or physical machine)

1. Transfer the installation package **tunnel-connector-install-scripts.tar.gz** to the virtual machine where you want to install the **Connector**.
2. Log in to the virtual machine and unpack the installation package.
3. Install a running container engine (either **Docker** or **Podman**) on the virtual machine.
4. Unpack **tunnel-connector-install-scripts.tar.gz**.
5. Change to the **tunnel-connector-install-scripts** directory.
6. Run the **./install-connector.sh** script to install the Connector on the virtual machine. Accept the license agreement when prompted to start the installation. Alternatively, you can accept the license by adding the **--accept-license true** parameter on the command line.
7. When the script finishes, verify that the Connector is installed by running either of the commands:

   ```
   docker ps
   ```

   ```
   podman ps
   ```

Installing more than one Secure Tunnel Connector to a host machine

If you want to install more than one Secure Tunnel Connector:

- Ensure they are from a different Secure Tunnel server or different IBM Cloud Pak for AIOps instance. If they are from the same Secure Tunnel server or IBM Cloud Pak for AIOps instance, you need to add another ApplicationMapping to one **Tunnel Connection** instead of creating another **Tunnel Connection** and installing another **Tunnel Connector**.
- Configure each Secure Tunnel operator to add the **applicationMappingPortRang: 30000-65000** item to the **YAML**. The **applicationMappingPortRang** should be different for each Secure Tunnel. For example, the first one should be **30000-40000** and the second should be **40001-50000**. Follow these steps to configure the **applicationMappingPortRang**:
  1. From the Red Hat OpenShift Container Platform console, click **Operators** > **Installed Operators**.
  2. From the **Project** menu, select the project in which you created the Secure Tunnel instance.
  3. Click the **IBM Secure Tunnel** > **Tunnel** tab > click the existing **sre-tunnel** instance > change to the **YAML** tab.
  4. Add **applicationMappingPortRang: 30000-65000** to the YAML and save it. The final YAML should look like:

```
spec:
  license:
    accept: true
  version: 1.0.0
  pullSecret: secure-tunnel-pull-secret
  showInContainerLog: true
```

```
runType: production
size: small
controllerReplicas: 1
applicationMappingPortRang: 30000-65000
```

The Secure Tunnel Connector container runs in the host network on host machines. Avoid port conflicts if you install multiple Secure Tunnel Connectors in the same host machine. For the following reasons, do not use port numbers **9000**, **50443** or **12443**:

- The Secure Tunnel Connector container listens on port **9000** for health check by default.
- The Secure Tunnel Connector container in a *private network to public network connector* setup connects to the Secure Tunnel server on port **50443** by default.
- Secure Tunnel Connector container for **Slack** and **Microsoft Teams** listens on port **12443** on **Exposed mapping port** by default.
- When you create a **Secure Tunnel Connection** from a **Custom** template and select **Secure Tunnel in private network initializes connection to connector in public network** in the **Direction** page, and select **Host machine** in the **Connector settings** page you must input a different port number than **50443** for your connection. And this port number must be different than the port number for all the connectors that run on the same host machine.

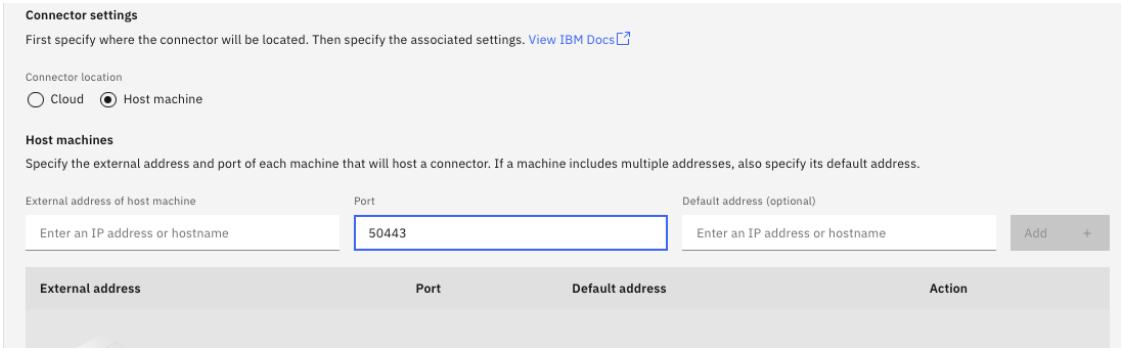


Figure. Port configuration

- When you create a **Secure Tunnel Connection** from the **ChatOps** template, and select **Host machine** in the **Connector settings** page you must input a different port number than **50443** in the **Port** field and a different port number to **12443** in **Exposed mapping port** field. These port numbers must be different than the port numbers for all the connectors you are running on the same host machine.

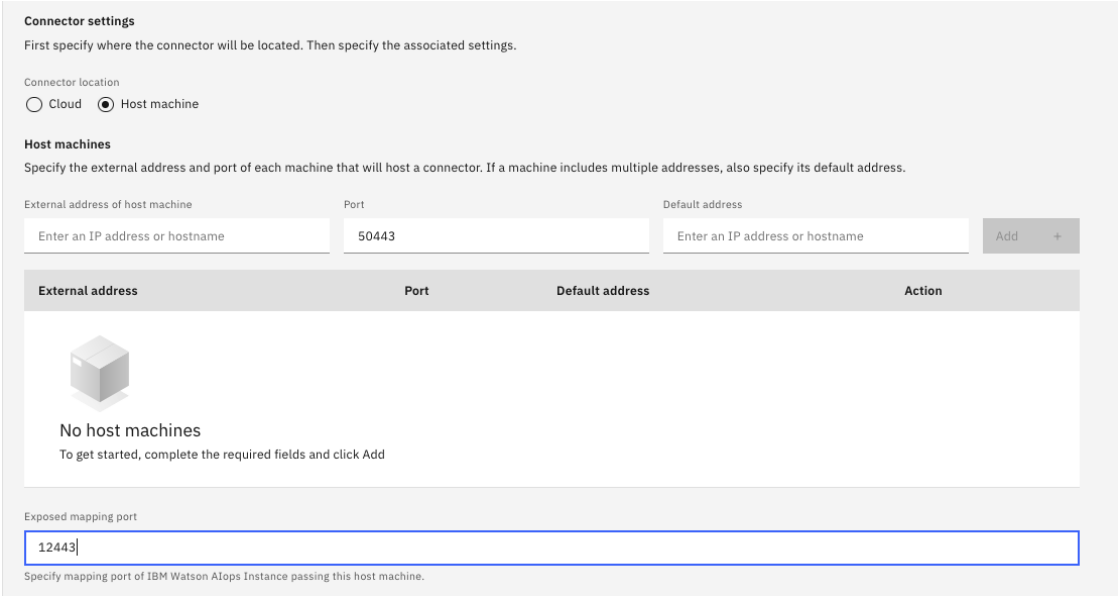


Figure. Port configuration

- When you complete creating the connection, you can download and transfer the installation package to the host machine on which a Secure Tunnel connector is already installed.

- When you install the connector, you must specify a different health check port to **9000** and the other health check ports that are used by other secure tunnel connectors that run on this host machine. Use the following command to assign a specific **<port number>**:

  ```
  ./install-connector.sh --accept-license true --port <port number>
  ```

Note: For Port conflicts **install-connector.sh** uses **fuser** to check port conflicts during installation. If you do not have **fuser** installed on the host machine, **install-connector.sh** might miss some conflicts. If it does, your connector container might fail to start.

You can check the Secure Tunnel Connector container logs by using either of the following commands:

```
docker logs <container id>
```

```
podman logs <container id>
```

Where **<container id>** is the ID of the container.

- If you find a port conflict on the health check port in the container log:
  1. Uninstall the container by using the command:

     ```
     ./uninstall-connector.sh
     ```

  2. Reinstall the connector with a different health check port. For example, `./install-connector.sh --accept-license true --port 9002`
- If you find a port conflict on the port that is used to connect the Secure Tunnel server or on the **Exposed mapping port**, then
  1. Uninstall the container by using the command:

     ```
     ./uninstall-connector.sh
     ```

  2. Delete the Secure Tunnel connection from the web UI.
  3. Create a new Secure Tunnel connection with different ports and try again.

For more information, see Troubleshooting Secure Tunnel.

Optional: Uninstalling the Secure Tunnel Connector

Use the script to uninstall the **Secure Tunnel Connector**:

1. Download the Connector installation package **tunnel-connector-install-scripts.tar.gz**
   1. From the console, go to **Manage connections**, locate the tunnel name.
   2. Under the **Action** column for the tunnel, expand the menu (three dots icon) and click **Install**.
   3. Obtain the installation package **tunnel-connector-install-scripts.tar.gz** by clicking **Download**. Or you can click **Copy**, and then run the copied commands on a Linux system.

      Note: The installation script is only for a specific **Secure Tunnel Connection**, and cannot be used for other Secure Tunnel Connections.
2. Uninstall the Connector from the **Cloud** with the installation package:
   1. Transfer the installation package **tunnel-connector-install-scripts.tar.gz** to a virtual machine on which the OpenShift CLI (**oc**) is installed. Log in and access the **Cloud** where you want to uninstall the Connector.
   2. Log in to the virtual machine and unpack the installation package.
   3. Change to **tunnel-connector-install-scripts** directory
   4. Run **./uninstall-connector.sh** the following script to uninstall the Connector from the Cloud:

      ```
      ./uninstall-connector.sh --namespace <namespace>
      ```

Where **<namespace>** is the project (namespace) where the Connector is installed.

3. Uninstall the Connector from the **host machine** with installation package:

    1. Transfer the installation package **tunnel-connector-install-scripts.tar.gz** to the host machine on which the Connector is installed.

    2. Log in to the virtual machine and unpack the installation package.

    3. Change to **tunnel-connector-install-scripts** directory.

    4. Run **./uninstall-connector.sh** the following script to uninstall the Connector on the host machine.

        ```
        ./uninstall-connector.sh --namespace <namespace>
        ```

        Where **<namespace>** is the project (namespace) the Connector is installed.

Optional: Delete the Secure Tunnel Connector resources after a tunnel connection is deleted

If you deleted a tunnel connection by clicking **Action** > **Delete** in the *Manage connections* page on the IBM Cloud Pak Automation console, and you did not uninstall the connector, you can delete the connector resources in the cluster virtual machine or the host machine manually. Otherwise, a new connector cannot be installed on the Cluster virtual machine or the host machine.

- If the connector is running in an Red Hat OpenShift Container Platform cluster or Red Hat OpenShift on IBM Cloud, follow the steps to delete connector resources:

    1. Log in to the Red Hat OpenShift Container Platform cluster or Red Hat OpenShift on IBM Cloud cluster from command line, and then find all connector resources by running the following command:

        ```
        oc get sts,secret,service,role,route -n <your tunnel connector namespace> \
              -l app.kubernetes.io/group=<your release name>-tunnel-connector.app
        ```

    2. You can find the **release name** by checking the tunnel server tunnel prefix of **tunnel-network-api**, like this, which gives the **release name** as **sre-tunnel**:

        ```
        oc get deploy -n cp4waiops | grep tunnel-network-api
            sre-tunnel-tunnel-network-api                                    3/3     3
        3           142m
        ```

    3. Remove these resources by running the following command:

        ```
        for res in $(oc get sts,secret,service,role,route -n <your tunnel connector
        namespace> \
              -l app.kubernetes.io/group=<your release name>-tunnel-connector.app -o name )
        do
        oc delete -n <your tunnel connector namespace> $res
        done
        ```

    4. If no other Tunnel connector is installed in the namespace, delete the secret by running the following command:

        ```
        oc get secret -n <your tunnel connector namespace> | grep tunnel-connector
        ```

        Note, Do not attempt to delete the secret if the tunnel connectors are still present as the secret might be used by one of the connectors.

Upgrading Secure Tunnel

Use the following instructions to help you upgrade Secure Tunnel either with IBM Cloud Pak for AIOps or as a standalone component.

Two methods are available to upgrade Secure Tunnel for use with IBM Cloud Pak for AIOps:

- Upgrading Secure Tunnel from IBM Cloud Pak for AIOps
- Upgrading Secure Tunnel as a stand-alone component in your Red Hat OpenShift cluster

Upgrading Secure Tunnel from IBM Cloud Pak for AIOps

Secure Tunnel is upgraded when you upgrade IBM Cloud Pak for AIOps. For details about upgrading IBM Cloud Pak for AIOps, see Upgrading.

Upgrading Secure Tunnel as a stand-alone component in your Red Hat OpenShift cluster

1. **Configure automatic catalog polling**

   - Ensure that your catalog is set to automatically poll for the latest images.
   - Your `ibm-operator-catalog` CatalogSource object can be configured to automatically poll for the latest catalog version, and to retrieve it if one is available. Polling for updates is enabled by configuring the polling attribute, `spec.updateStrategy.registryPoll`.
   - You might have elected to automatically accept updates by adding the polling attribute to your `ibm-operator-catalog` YAML when you installed Secure Tunnel, when you installed a Secure Tunnel hotfix from IBM support Opens in a new tab, or when you installed another IBM Cloud Pak.
   - Use the following steps to check whether you have a polling attribute set. If you do not have one, configure a polling attribute set:
     - Log in to your OpenShift cluster's console.
     - Go to **Administration** > **Cluster Settings**.
     - Click **Configuration**, scroll down and click **OperatorHub**.
     - Select **Sources**, scroll down and click `ibm-operator-catalog`.
     - Click **YAML** to switch to the YAML view. If there is no `spec.updateStrategy` section, or `spec.image` is not set to `icr.io/cpopen/ibm-operator-catalog:latest`, then update the YAML to have the following entry in the `spec` block, and save it.

       ```
       spec:
         displayName: ibm-operator-catalog
         publisher: IBM Content
         sourceType: grpc
         image: icr.io/cpopen/ibm-operator-catalog:latest
         updateStrategy:
           registryPoll:
             interval: 45m
       ```

       Note: `ibm-operator-catalog` also contains the catalogs for other IBM Cloud Paks. If you have multiple IBM Cloud Paks installed on your cluster and you enable the polling attribute, then automatic update is configured for all of them.

2. **Update the operator subscription**

   Use the following steps to update the `spec.channel` value of the Secure Tunnel subscription to the release that you want to upgrade to:

   1. Log in to your OpenShift cluster's console.
   2. Select **Home** > **Search**.
   3. From the **Project** list, select the project (namespace) that your Secure Tunnel subscription is deployed in. This is your Secure Tunnel project if your deployment is namespace scoped, or openshift-operators if your deployment has a cluster wide scope.
   4. In the **Resources** list, select **SUB Subscription**. A list of subscriptions is displayed.
   5. Click the subscription that has the name `ibm-secure-tunnel-operator`. A new window with the subscription details for `ibm-secure-tunnel-operator` is displayed.
   6. Click the value in the **Update channel** box. A new window called *Change Subscription update channel* is displayed.
   7. Change the channel to the version that you want and click **Save**. If the subscription has been configured with **Manual** for the **Update approval**, then you need to click **Upgrade available** on the **Upgrade status** > **Preview**

InstallPlan > **Approve** to approve this upgrade.

Note: You need to upgrade the channel versions one by one. For example, if your current channel version is 3.7.x, you cannot skip the 4.1.x to change the channel to 4.2.x. You must upgrade it to 4.1.x and then upgrade it to 4.2.x.

3. **Verify the deployment**

   Use the following procedure to check the status of your upgraded Secure Tunnel deployment. It takes around 2-10 minutes for the upgrade to complete (subject to the speed with which images can be pulled).

   1. Log in to your OpenShift console.
   2. Click **Operators** > **Installed Operators**.
   3. From the **Project** list, select the project (namespace) that Secure Tunnel is deployed in if your deployment is namespace scoped, or openshift-operators if your deployment has a cluster-wide scope.
   4. Locate **IBM Secure Tunnel** in the list and verify that the annotation shows the version that is required.

Creating Secure Tunnel connections

A Secure Tunnel connection links the network that contains the Secure Tunnel to a remote network. Your applications can be on either side of the connection.

Secure Tunnel comes with connection templates for common application scenarios, including ChatOps, VMware, and Turbonomic. You can also create custom connections to support more scenarios.

- Prerequisites
- Secure Tunnel Web Console
- Creating a secure tunnel
  - ChatOps connection
    - ChatOps connection (Microsoft Azure as a host machine)
- VMware connection
- Turbonomic connection
- OpenStack connection
- Custom connection
  - Creating a Custom Instana connection

Prerequisites

- Ensure that you have Red Hat OpenShift Container Platform installed on one of your networks (referred to as **network-A** in the following steps). You use this instance of OpenShift to bridge to the other network.
- Ensure that another system (Red Hat OpenShift Container Platform, Kubernetes, or other system) or a host machine (virtual machine or physical machine) is available on a different network (referred to as **network-B** in the following steps). You use this network as the location for installing the Secure Tunnel Connector.
- Ensure that the network access is available on **network-A** and **network-B**. Either **network-A** or **network-B** needs to have a public IP address.
- Ensure that you install Secure Tunnel. For more information, see Installing Secure Tunnel.
- Verify that you can access the Secure Tunnel web console.

Secure Tunnel UI

Use either of these methods to access the Secure Tunnel console:

1. If you installed Secure Tunnel in IBM Cloud Pak for AIOps,login to the IBM Cloud Pak Automation console. From the navigation menu(four horizontal bars), click **Administration** > **Secure Tunnel**.

2. If Secure Tunnel is not installed in IBM Cloud Pak for AIOps, use the Red Hat OpenShift Container Platform route to access the stand-alone Secure Tunnel. To get the Secure Tunnel URL, enter the following command:

   ```
   oc get route -n <namespace> sre-tunnel-tunnel-ui -o jsonpath='{.spec.host}'
   ```

Where **<namespace>** is the project (namespace) where Secure Tunnel is installed.

With the Secure Tunnel UI, you can create the following types of connections:

- ChatOps connection
- VMware connection
- Turbonomic connection
- OpenStack connection
- Custom connection

Creating a Secure Tunnel

ChatOps connection

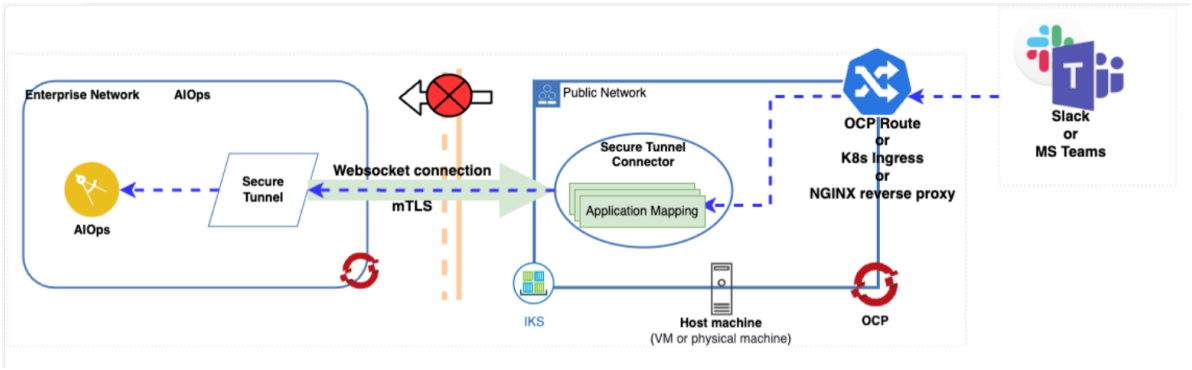


Figure. High-level design for VMware

To integrate IBM Cloud Pak for AIOps with a service like Slack or Microsoft Teams, the IBM Cloud Pak for AIOps instance needs to be accessible from the public network. If your IBM Cloud Pak for AIOps is not accessible from the public network, you can open the IBM Cloud Pak for AIOps instance to the public network by creating a Secure Tunnel. You can do it by exposing the IBM Cloud Pak for AIOps instance to either a cluster (Red Hat OpenShift Container Platform or IBM Kubernetes Service cluster) or a host machine (virtual machine or physical machine), which is accessible from the public network. Simplify the creation of a secure tunnel by using the ChatOps template from the Secure Tunnel Console on the **Manage connections** page. Take the following steps to create a ChatOps connection:

1. Click the **ChatOps** tile and a panel with an introduction to the ChatOps template opens. Click **Get started**.
2. In the *Name and Tags* section, provide a name and tag for this connection.
    - *Connection name*: required. The name of the connection. The name can include only lowercase alphanumeric characters and '-', and must start and end with a lowercase alphanumeric character.
    - *New tag*: optional. The tag of the Connection.
3. In the *Connector settings* section:

    Specify where the connector is located, such as whether it is a Cloud or Host machine.

    - If you choose to install the connector in the Red Hat OpenShift cluster, the connector can be installed automatically. Complete the following actions:
        - *Login command for target Red Hat OpenShift cluster*: required. Specify a publicly accessible cluster with a CA-signed domain certificate. Do not install the connector in the cluster that contains the Tunnel Connection worker.
        - *Target namespace to install the connector*: required. The project (namespace) where you want to install the connector.
        - *Connector replicas*: configure the tunnel connector replicas to set High Availability.
    - If you want to install the connector to Red Hat OpenShift Container Platform manually, you can use the Custom method
    - If you choose to install the connector in a Kubernetes cluster, complete the following actions:

- *Cloud domain*: Run the command to get IBM Kubernetes Service cluster domain name:

  ```
  ibmcloud ks cluster get --cluster <Cluster ID> --output json | jq
  '.ingress.hostname'
  ```

- *Target namespace to install the connector*: required. The project (namespace) where you want to install the connector.

- *Connector replicas*: you will config the tunnel connector replicas to set High Availability.

- If you choose to install the connector in the Host machine, complete the following actions:

  - *Host machine addresses*: required. Includes the external address of the host machine, Port and default address of the host machine. When a host machine includes multiple addresses, a default IP address is required. If you want to use one host machine to expose more than one IBM Cloud Pak for AIOps, you need to set a different port for other ChatOps connections.

    - *Expose mapping port*: required. Default is `12443`. You can change this port to allow several IBM Cloud Pak for AIOps instances to have different exposed ports. Then, you can use one host machine to create more than one ChatOps connection.

  - *DNS and load balancing*: required. First, enter the DNS name of a suitable host machine. Next, select the same machine to host a reverse proxy / load balancer (bundled with the connector). Finally, specify the corresponding certificate and private key.

    Note: If you have an IBM Cloud account, see Getting started with IBM Cloud Internet Services to apply for a **DNS name**, and see Preparing to order public certificates to apply for a CA-signed certificate. Extract the CA-signed certificate file, `hostname.pem`, for the certificate and `hostname.key` for the private key. If an intermediate file is in your CA-signed certificate file – such as hostname-intermediate.pem – concatenate this file with a cert file, such as hostname.pem, by running the following command:

    ```
    ```sh {: codeblock}
    cat hostname-intermediate.pem >> hostname.pem
    ```
    ```

4. In the *Secure Tunnel settings*, configure the tunnel worker replicas to set High Availability.

5. *HTTPS proxy*: optional. Specify a forward proxy to enable the Secure Tunnel to reach the connector in the other network, such as username:password@hostname:port.

   Note: It is necessary to enter the HTTPS proxy information when Secure Tunnel is in an air-gapped environment. If your Secure Tunnel is in an air-gapped environment without internet connectivity, you need to prepare a host machine (virtual machine or physical machine). Ensure that the host machine can be accessed from the IBM Cloud Pak for AIOps cluster, and host machine can also connect to the internet. Then, install an HTTPS proxy server in the host machine, and set the HTTPS proxy server in the HTTPS proxy field.

   If you install a connector in Red Hat OpenShift Container Platform, and the Secure Tunnel in an air-gapped environment and you configured the HTTPS proxy for it, choose **Kubernetes** for the **Cloud Type** in the above step. Then, to get the domain for the **Cloud domain** field, run the command:

   ```
   HOSTNAME=$(oc get route -n openshift-console console -o=jsonpath={.spec.host}) &&
   HOSTNAME=${HOSTNAME#console-openshift-console.} && echo $HOSTNAME
   ```

   For example, squid is a simple HTTPS proxy server. See the following example of using squid to set up an HTTPS proxy server (Do not use this example in your production environment). For a production environment, contact your network administrator to help you configure the HTTPS proxy.

   Save the following configurations in a file `squid.conf`:

   ```
   acl all src 0.0.0.0/0.0.0.0
   acl SSL_ports port 443
   acl CONNECT method CONNECT

   http_access deny CONNECT !SSL_ports

   http_access allow all

   # Squid normally listens to port 3128
   ```

```
http_port 3128

# Uncomment and adjust the following to add a disk cache directory.
cache_dir ufs /var/spool/squid 100 16 256
# Leave coredumps in the first cache dir
coredump_dir /var/spool/squid
#
# Add any of your own refresh_pattern entries above these.
#
refresh_pattern ^ftp:         1440    20%    10080
refresh_pattern ^gopher:    1440    0%    1440
refresh_pattern -i (/cgi-bin/|\?) 0    0%    0
refresh_pattern (Release|Packages(.gz)*)$      0       20%     2880
# example lin deb packages
#refresh_pattern (\.deb|\.udeb)$   129600 100% 129600
refresh_pattern .         0    20%    4320
```

Start the squid server by running the following command:

```
docker run --name squid -d -p 3128:3128 -v `pwd`/squid.conf:/etc/squid/squid.conf
datadog/squid
```

Set the `<host machine IP>:3128` in the **HTTPS proxy** field.

6. Click **Create connection** to create a ChatOps connection.

   - If you choose to install the connector in the Red Hat OpenShift cluster, the connector is automatically installed. You can see the installation status in the connector installation section. When the installation is finished, you can get the application mapping URLs for Slack integration and Microsoft Teams integration.

     Note: If the installation fails, wait for about 30 seconds to see if it retries. If the installation does not succeed, you can reinstall the connector manually or re-create the ChatOps connection.

   - If you choose to install the connector in Kubernetes or a host machine, the connector must be manually installed. You can get the installation script in the *Next steps* section. Install the connector by following this installation guide. When the connector is installed, you can use the appropriate application mapping address to integrate with Slack or Microsoft Teams.

ChatOps connection (Microsoft Azure as host machine)

To set up a ChatOps connection with Microsoft Azure as the host machine, you need to install IBM Cloud Pak for AIOps in network-A (private cloud) and set up ChatOps in network-B (Slack or Microsoft Teams in public cloud). This process involves configuring ChatOps and a Secure Tunnel out of network-A to communicate with ChatOps in the public cloud, network-B.

1. Install IBM Cloud Pak for AIOps. For more information, see Installing

2. Deploy a VM to Azure, and give it a public internet-facing endpoint. **Note:** This must have a signed trusted certificate that ChatOps accepts as trusted (that is not self-signed).

3. On Azure, open the `50443` port on the VM that uses an Network Security Group (NSG).

4. Give the VM a public DNS name.

5. Check whether it is a private IP on the Azure UI. For example, the private IP in this case is `172.18.0.4`:

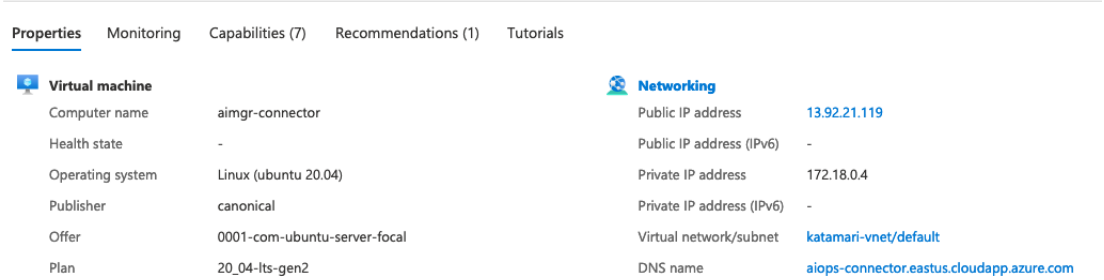


Figure. Azure UI

6. Get the signed trusted certs, including the root and any intermediate CAs.

7. To configure the tunnel connector on the Secure Tunnel console, enter it as shown here:

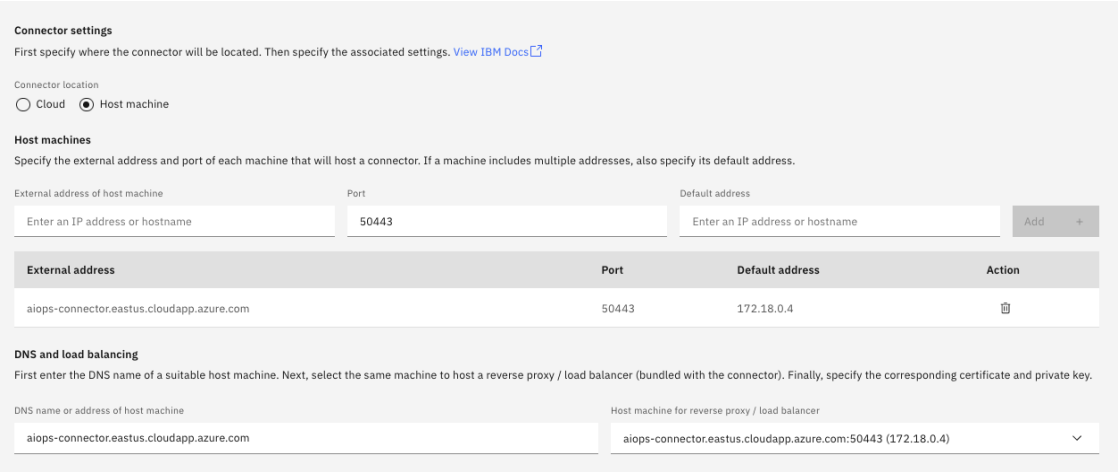


Figure. Secure Tunnel UI

8. Note where the private IP is entered and note the values for DNS. You need to select the host machine in the dropdown menu.

9. Enter your cert and keys in the connector UI.

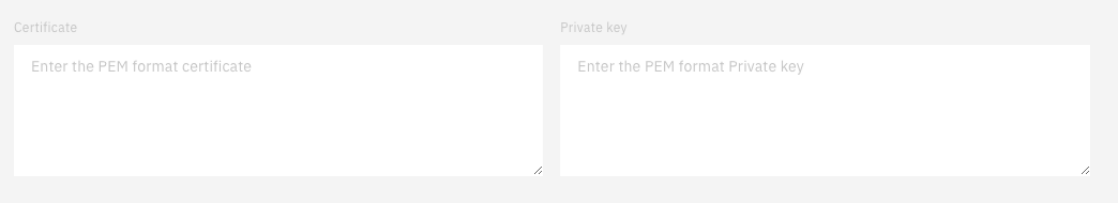


Figure. Enter certifaction keys

10. You then get a **`tar.gz`**. Download this file and transfer it to Azure VM.

11. On the Azure VM, where you unpack the download, ensure that nothing is connected to the **`443`** port (like an Apache server) and run the secure tunnel installer as root, as it needs that privilege to bind to the **`443`** port.

12. You then get a successful installation and see a *Ready* status back on the Secure Tunnel UI.

    Script output

```
Install Tunnel Connector successfully
{"ok":true}
{"ok":true}

Start Tunnel Connector successfully

{"description":""}
{"description":""}
A
Tunnel Connector is ready to use.
```

    IBM Cloud Pak for AIOps UI

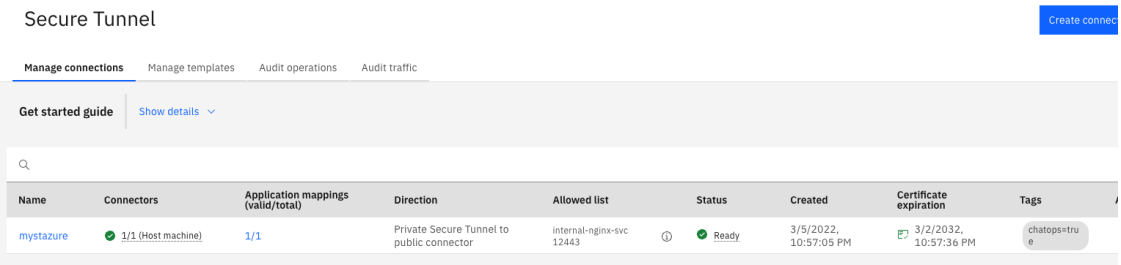


Figure. Additional Secure Tunnel UI

13. Click the application mappings entry, which shows you the URLs to use in your Slack or Microsoft Teams. You can renew the certificate and private key by editing the application mapping.

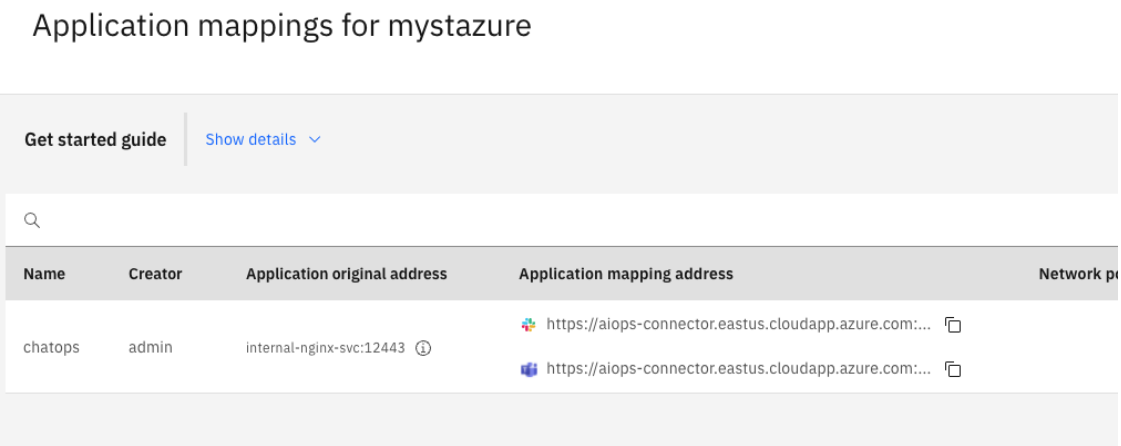


Figure. Application mappings

14. In the Microsoft Teams bot, enter the URL in the **Messaging endpoint** field.

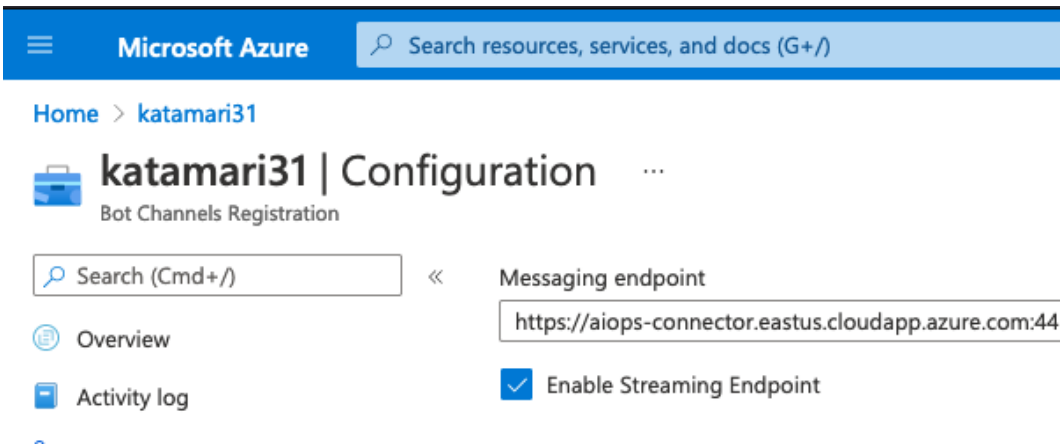


Figure. Azure team

VMware connection

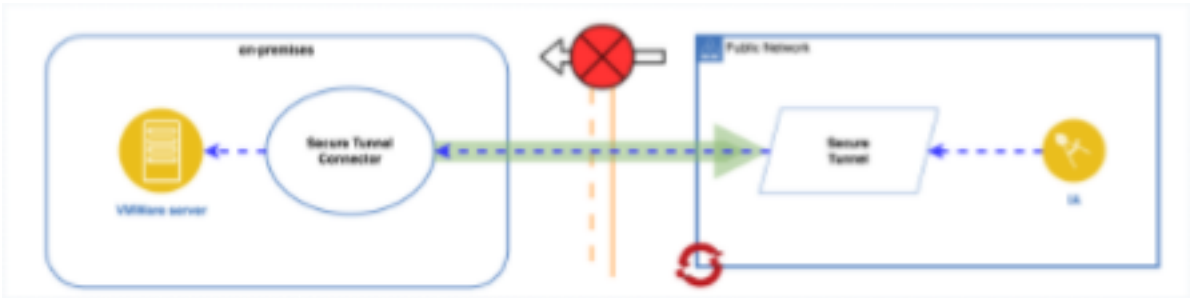


Figure. High-level design for VMware

Follow the installation instructions to install Secure Tunnel.

You can access services in a private VMware® vCenter from a public cloud **Infrastructure automation**. In this case, install a Secure Tunnel in the Infrastructure automation cluster, with IBM Cloud Pak for AIOps or a stand-alone version. Then, take the following steps to configure the VMware tunnel connection:

Note: VMware template supports the VMware scenario specifically. Other scenarios can be created by using a Custom template.

Simplifying the creation of a Secure Tunnel by using VMware template in IBM Cloud Pak for AIOps:

1. Click **VMware** tile. The VMware side panel opens and provides an introductory explanation to the VMware template. Click **Get started**.

2. In the *Name and Tags* section, provide a name and tag for this connection:

   - Connection name: required. The name of the connection. The name can include only lowercase alphanumeric characters and '-', and must start and end with a lowercase alphanumeric character.
   - New tag: optional. The tag of the Connection.

3. In the *Connector settings* section, configure the tunnel connector replicas to High Availability and install the tunnel connector to Cloud or Host machine by selecting the connector location section.

4. In the *Secure Tunnel setting*, configure the tunnel worker replicas to set High Availability.

5. *HTTPS proxy*: optional. Specify a forward proxy to enable the Secure Tunnel to reach the connector in the other network, such as username:password@hostname:port.

   Note: It is necessary to enter the HTTPS proxy information when Secure Tunnel is in an air-gapped environment. If your Secure Tunnel is in an air-gapped environment without internet connectivity, you need to prepare a host machine (virtual machine or physical machine). Ensure that the host machine can be accessed from the IBM Cloud Pak for AIOps cluster, and can also connect to the internet. Then, install an HTTPS proxy server in the host machine, and set the HTTPS proxy server in the HTTPS proxy field.

   If you install the connector in OCP, and Secure Tunnel in an air-gapped environment, you need use the Custom method to create a ChatOps connection

   For example, squid is a simple HTTPS proxy server. See the following example of using squid to set up an HTTPS proxy server. (It is just an example. Don't use it in your production environment.) For the production environment, contact your network administrator to help you configure the HTTPS proxy.

   1. Save the following configurations in a file squid.conf.

      ```
      acl all src 0.0.0.0/0.0.0.0
      acl SSL_ports port 443
      acl CONNECT method CONNECT

      http_access deny CONNECT !SSL_ports

      http_access allow all

      # Squid normally listens to port 3128
      http_port 3128

      # Uncomment and adjust the following to add a disk cache directory.
      cache_dir ufs /var/spool/squid 100 16 256
      # Leave coredumps in the first cache dir
      coredump_dir /var/spool/squid
      #
      # Add any of your own refresh_pattern entries above these.
      #
      refresh_pattern ^ftp:        1440    20%    10080
      refresh_pattern ^gopher:    1440    0%    1440
      refresh_pattern -i (/cgi-bin/|\?) 0    0%    0
      refresh_pattern (Release|Packages(.gz)*)$      0       20%     2880
      # example lin deb packages
      #refresh_pattern (\.deb|\.udeb)$   129600 100% 129600
      refresh_pattern .        0    20%    4320
      ```

   2. Start the squid server by running the following command:

      ```
      docker run --name squid -d -p 3128:3128 -v `pwd`/squid.conf:/etc/squid/squid.conf
      datadog/squid
      ```

   3. Set the < host machine IP >:3128 in the **HTTPS proxy** field.

6. In the *Application mappings* section, edit the VMware addresses to match your environment. If it is the same as the address in the template, you don't need to change anything.

7. Click **Create connection** to create a VMware connection. You can get the installation script in the *Next steps* section. Install the Connector by following the guide. When the connector is installed, the application mapping address is accessible. You can use this address as the VMware vCenter service address to configure the VMware connection from the Infrastructure Automation UI.

Turbonomic connection

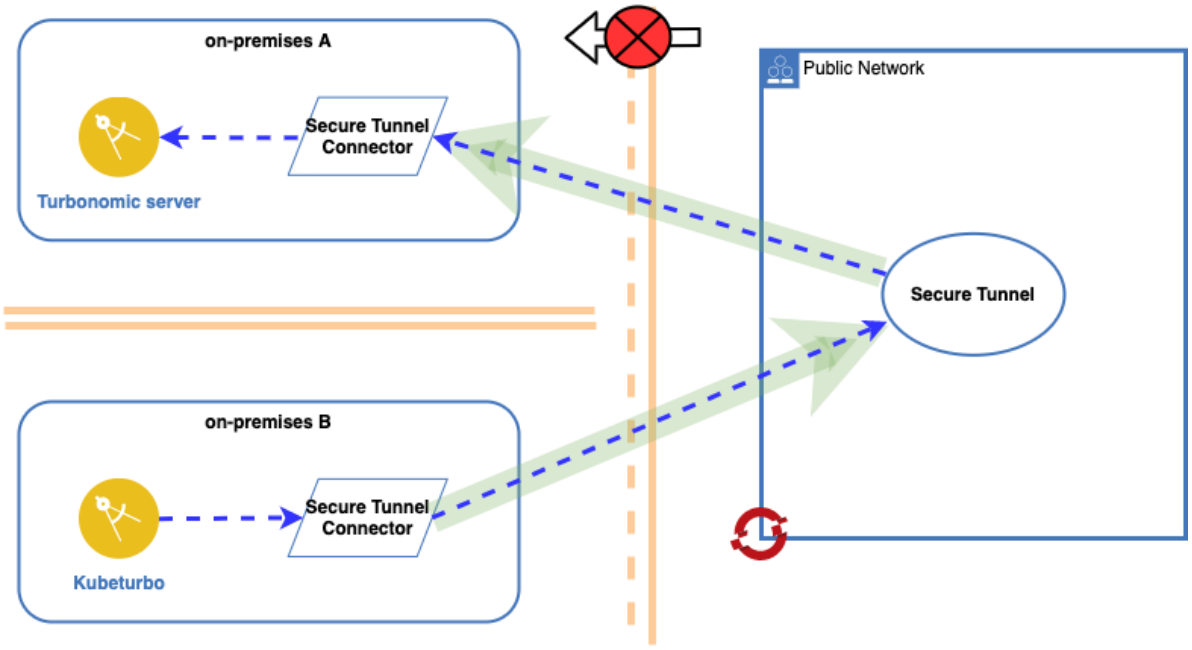


Figure. High-level design for Turbonomic

The Turbonomic server is in one private network cluster and you want to integrate it with a kubeturbo probe in another private network.

In this case, install a stand-alone Secure Tunnel on a public Cloud. Then, create one tunnel connection to export the **Turbonomic** server endpoint to the public cloud cluster. Next, create another tunnel connection to export the exposed **Turbonomic** server endpoint (a Kubernetes service in the public cloud cluster) to the **kubeturbo** private network cluster. The **kubeturbo** can now access the **Turbonomic** server through the Secure Tunnel.

Simplifying the creation of a secure tunnel by using the Turbonomic template:

1. Click the **Turbonomic** tile to open the Turbonomic side panel. Next, click **Get started** to create a connection. The first connection exposes Turbonomic to the Secure Tunnel server.
2. In the **Name and Tags** section, provide a name and tags for this connection:
   - *Connection name*: required. The name of the connection. The name can include only lowercase alphanumeric characters and '-', and must start and end with a lowercase alphanumeric character.
   - *New tag*: optional. The tag of the Connection.
3. In the *High availability* section:
   - *Connector replicas*: configure the tunnel connector replicas to High Availability.
   - *Tunnel worker replicas*: configure the tunnel connector replicas to High Availability.
4. In the *Application mappings* section, edit the Turbonomic server addresses to match your environment.(**In particular, check the namespace part of the Turbonomic Kubernetes service address, and check whether it matches the Turbonomic Kubernetes service address**). If it is the same as the address in the template, you don't need to change anything.
5. Click **Create connection** to create a Turbonomic connection to expose the Turbonomic server to the public cloud cluster (as a Kubernetes network service), then you get the installation script in the *Next steps* section. You can install the connector by following this guide or install it later in the *Manage connections* page, by clicking **Action** > **installation connector**.

6. Click **Next**. Install the second connection. This step is the same as the first. The difference is that you cannot change the application address in the *Application mappings* section. Now create a tunnel connection to export the exposed **Turbonomic** server from the public cluster (a Kubernetes network service) to the **kubeturbo** private network cluster. Then, click **Create connection** to create a Turbonomic connection. You can get the installation script in the **Next steps** section, and install the connector.

7. Finally, select the **`<the name that you provided in the Name and Tags>-kubeturbo`** tunnel connection and obtain the application mapping address. Then, you can use this application mapping address as the Turbonomic server address in the **kubeturbo**. The address is similar to this: **`http://sre-tunnel-9a4f8fbb26374fa7.tunnel-connector-turbo:8080`**.

8. Install the **kubeturbo**. Create a Project(namespace), like **`kubeturbo`**, to install the **kubeturbo** operator and instance from the Red Hat OpenShift console **OperatorHub**. Enter **`kubeturbo`** into the **All Items** field. Click the **Kubeturbo Operator** panel to install the **kubeturbo**. After the operator is installed, navigate to **Installed Operators** > **Kubeturbo Operator**, and click **Create instance** from the **Kubeturbo operator** panel. Change to the **YAML View** and edit the YAML like this:

```
spec:
  replicaCount: 1
  restAPIConfig:
    opsManagerPassword: <password of the Turbonomic server>
    opsManagerUserName: <username of the Turbonomic server>
  serverMeta:
    turboServer: '<the turbo server application mapping address, should be like:
http://sre-tunnel-9a4f8fbb26374fa7.tunnel-connector-turbo:8080>'
    version: '8'
  targetConfig:
    targetName: <give a name like: tunneltest>
```

Note: Check that the connectors have successfully installed for the two connections. One is in the private Turbonomic server cluster, the other is in the private kubeturbo agent container.

OpenStack connection

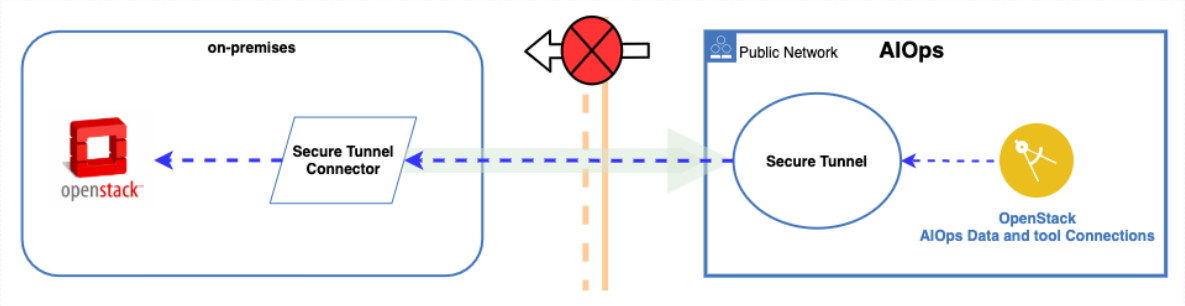


Figure. High-level design for Openstack

Scenario: The OpenStack server is in a private network cluster and IBM Cloud Pak for AIOps is in a public cluster. To integrate with the private OpenStack server, you can use a Secure Tunnel to create an OpenStack connection. The OpenStack template that is described in the following section supports this type of connection. You can create other templates through a custom connection. For more information, see Custom connection.

Simplifying the creation of a Secure Tunnel by using an OpenStack template:

From the **Secure Tunnel Console** on the **Manage connections** tab page, click **Create connection**. Then, take the following steps to create an OpenStack connection:

1. Click the **OpenStack** tile. The OpenStack introductory side panel opens. Click **Get started**.

2. In the **Name and Tags** section, provide a name and tags for this connection:

   - *Connection name*: required. The name of the connection. The name can include lowercase alphanumeric characters and '-', and must start and end with a lowercase alphanumeric character.
   - *New tag*: optional. A tag for the connection.

3. In the **Connector settings** section, configure the tunnel connector replicas to set High Availability and choose to install the tunnel connector to **Cloud** or **Host machine** by selecting the **Connector location** section.

4. In the *Secure Tunnel setting*, configure the tunnel worker replicas to set High Availability.

- In the *HTTPS proxy*, optionally specify a forward proxy to enable the Secure Tunnel to reach the connector in the other network, such as *username:password@hostname:port*.

  Note: It is necessary to enter the HTTPS proxy information when Secure Tunnel is in an air-gapped environment. If your Secure Tunnel is in an air-gapped environment without internet connectivity, you need to prepare a host machine (virtual machine or physical machine). Ensure that the host machine can be accessed from the {site.data.keyword.cp4waiops_notm}} cluster, and can also connect to the internet. Then, install an HTTPS proxy server in the host machine and set the HTTPS proxy server in the HTTPS proxy field.

  For example, squid is a simple HTTPS proxy server. See the following example of using squid to set up a HTTPS proxy server. Do not use it in your production environment. For the production environment, contact your network administrator to help you configure the HTTPS proxy.

  1. Save the following configurations in a file squid.conf.

     ```
     acl all src 0.0.0.0/0.0.0.0
     acl SSL_ports port 443
     acl CONNECT method CONNECT

     http_access deny CONNECT !SSL_ports

     http_access allow all

     # Squid normally listens to port 3128
     http_port 3128

     # Uncomment and adjust the following to add a disk cache directory.
     cache_dir ufs /var/spool/squid 100 16 256
     # Leave coredumps in the first cache dir
     coredump_dir /var/spool/squid
     #
     # Add any of your own refresh_pattern entries above these.
     #
     refresh_pattern ^ftp:         1440    20%    10080
     refresh_pattern ^gopher:    1440    0%    1440
     refresh_pattern -i (/cgi-bin/|\?) 0    0%    0
     refresh_pattern (Release|Packages(.gz)*)$      0       20%     2880
     # example lin deb packages
     #refresh_pattern (\.deb|\.udeb)$   129600 100% 129600
     refresh_pattern .         0    20%    4320
     ```

  2. Start the squid server by running the following command:

     ```
     docker run --name squid -d -p 3128:3128 -v `pwd`/squid.conf:/etc/squid/squid.conf
     datadog/squid
     ```

  3. Set the :3128 in the HTTPS proxy field.

6. In the *Application mappings* section, edit the OpenStack addresses to match your environment, and enable *Origin application uses HTTPS*.

7. Click **Create connection** to create an OpenStack connection. You can get the installation script in the **Next steps** section. Follow the installing connector steps. After installing the connector, the application mapping address is accessible. Use this address as the VMware vCenter service address to configure the VMware connection from the Infrastructure Automation UI.

Custom connection

Secure Tunnel can accommodate specific requirements with a Custom connection. For instance, you can use this type of connection for ChatOps to help you understand how to customize your own connection.

The installation connector in Red Hat OpenShift is automatically supported when you use the ChatOps template. If you do not prefer this approach, you can install the connector in Red Hat OpenShift manually with the Custom method.

Use case summary: IBM Cloud Pak for AIOps is installed. Slack (for ChatOps) is in a public network. To access IBM Cloud Pak for AIOps, expose the IBM Cloud Pak for AIOps instance to the public cluster through a connector that is installed in Red Hat

OpenShift.

1. Create a **connection** to create a bridge configuration for the network where **Secure Tunnel** is installed and the network that has the connector.

    1. Click **Custom** tile. This opens the Custom side panel. Then, click **Get started**.

    2. In the *Name and Tags* section, Provide a name and tags for this connection:

        - Connection name: required. The name of the connection. The name can include lowercase alphanumeric characters and '-', and must start and end with a lowercase alphanumeric character.

        - New tag: optional. The tag of the Connection. Add the tag **`chatops=true`** in our example.

    3. In *Direction*, specify whether the relationship is one of the below:

        - *connector in private network initializes connection to Secure Tunnel in public network*.
        - *Secure Tunnel in private network initializes connection to connector in public network*.

        Because Secure Tunnel is in the IBM Cloud Pak for AIOps cluster, you need to choose the option **Secure Tunnel in private network initializes connection to connector in public network**.

    4. In the *Connector settings* section:

        1. *Connector location*: required. Select whether installation connector is in the Cloud or in a Host machine. Because the connector is installed on Red Hat OpenShift, choose **Cloud**.

        2. *Connector replicas*: Configure the tunnel connector replicas to set High Availability, 1 - 16, as you need.

        3. *Cloud domain*: If you install the connector in the Cloud direction, the Secure Tunnel in the private network initializes the connection to a connector in a public network. You need to input the cloud domain of the cluster where you installed the connection. Run the following command to get the Red Hat OpenShift cluster domain name:

            ```
            oc get route -n openshift-console console -o jsonpath={.spec.host} | sed -e 's/[^\.]*\.//
            ```

            Run the following command to get the IBM Kubernetes Service cluster domain name:

            ```
            ibmcloud ks cluster get --cluster <Cluster ID> --output json | jq '.ingress.hostname'`
            ```

        4. *Host Machine*: if you install the connector in a VM or Physical machine, you need:

            1. Host machine addresses
                1. If your Connection direction is *Secure Tunnel in private network initializes connection to connector in public network*, it must include the *External address of host machine*, *Port* and *Default address of host machine* .(Optionally, specify a default IP address when a host machine includes multiple addresses).
                2. If your connection direction is *Connector in private network initializes connection to Secure Tunnel in public network*, it must include *Default address of host machine* (optionally, specify a default IP address when a host machine includes multiple addresses.)
                3. *Common DNS name for all inbound traffic (optional)*: required, in this case. Specify a DNS name for all inbound (connector to tunnel) traffic. When multiple host machines are present, you can enter a DNS name or the external IP address or hostname of the machine hosting the reverse proxy/load balancer. **Note:** do not complete this field unless you have at least one host machine address.
                4. *Host machine for reverse proxy / load balancer (optional)*: required, in this case. All inbound (connector to tunnel) traffic is routed to an NGINX-based reverse proxy/load balancer deployed to this host machine during installation of the connector.
        5. *Connector replicas*: configure the tunnel connector replicas to set High Availability.

    5. In the *Secure Tunnel setting*, configure the tunnel worker replicas to set High Availability, limit: 1-16.

    6. *HTTPS proxy*: optional. Specify a forward proxy to enable the Secure Tunnel to reach the connector in the other network, such as username:password@hostname:port.

Note: It is necessary to enter the HTTPS proxy information when Secure Tunnel is in an air-gapped environment. If your Secure Tunnel is in an air-gapped environment without internet connectivity, you need to prepare a host machine (virtual machine or physical machine), and ensure that the host machine can be accessed from the IBM Cloud Pak for AIOps cluster and can also connect to the internet. Then, install a HTTPS proxy server in the host machine, and set the HTTPS proxy server in the HTTPS proxy field.

7. *Allowed list*: Specify the IP addresses and ports that are exposed across this connection through application mappings. You can specify address ranges and port ranges. If you do not define an allowed list, the application mapping URL is inaccessible. To expose the IBM Cloud Pak for AIOps instance, input IBM Cloud Pak for AIOps instance address: **`internal-nginx-svc 12443`**.

8. Click **Create connection** to initiate a custom connection. In the *Next steps* page, you install a connector and create application mappings.

2. Follow the Install Secure Tunnel Connector guide to enable the connection.

3. Create *Application Mapping* to expose the applications to another network (Secure Tunnel side to connector side, or opposite). Set application mappings to expose the IBM Cloud Pak for AIOps instance. Go to the **Secure Tunnel Console** on the *Manage connections* tab page, and click the **Application mappings** of the *tunnel connection* list. You are redirected to the *Application mappings* page.

   You have two ways to create a connection:

   - *Create a connection from scratch*
   - *Create a connection from existing template*
   - Create a connection from scratch

     - Click **Create** to open the wizard panel to create an application mapping from scratch.

     - In the *Name and Tags* section, provide a name and tags for this application mapping:

       - Application mapping name: required. The name of the application mapping name can include lowercase alphanumeric characters and '-', and must start and end with a lowercase alphanumeric character.
       - New tag: optional. The tag of the application. Add a tag **`chatops=true`** in our example.

     - In *Direction*, specify how you expose the application *from connector side to Secure Tunnel side* or *from Secure Tunnel side to connector side*. Because the IBM Cloud Pak for AIOps instance is in Secure Tunnel side, in this case, choose **from Secure Tunnel side to connector side**.

     - In the *Application details* section, specify the address of the application and the ports that you want to expose.

       The address and ports must be included in the allowed list for the connection. Configure DNS so that the application can be accessed by using the original address or the application mapping address (optional). Input the IBM Cloud Pak for AIOps instance address: **`internal-nginx-svc`**, and port: **`12443`**, and the exposed mapping port (default is **`12443`**. Change as you need.). **Note:** Before you add an IP address or hostname, check the existing application mappings from the *Application mappings* page in the IBM Cloud Pak Automation console. Check for an existing application mapping. If one is available, check whether the original address is the same IP address or hostname as the new one that you want to specify. Edit the existing mapping to add the new port value instead of creating another application mapping.

     - Configure the *DNS* so that application can be accessed by using the original address or application mapping address.

     - If you choose to expose the application *from the connector side to the Secure Tunnel side* in the *Direction* section, you can optionally configure the **Reverse proxy**. Toggle each switch to *Yes* to enable the following:

       - *Origin application uses HTTP/HTTPS, and server will check hostname or respond to other URLs of application through HTTP response payload* (for example, Falcon LogScale)
       - *Origin application uses HTTPS* (for example, OpenStack)
       - *Origin application responds to other URLs of application through HTTP response payload* (for example, OpenStack)

     - In *Access rule*, you can use access rules to control the traffic flow at the IP address or port level for particular applications in your cluster. The default network policy allows all ingress and egress traffic for

each application mapping. *Network policy*: Network policies help you specify how a pod is allowed to communicate with various network "entities" over the network. Network policies apply to a connection with a pod on one or both ends, and are not relevant to other connections.

- Create a connection from existing template:
  - Click **Copy from template** button, and the panel opens.
  - Choose a *template*. You can see the application mappings in this template is displayed in the table.
  - In the table, you can edit the *Application Mapping name*, *Application original address*, *Application original Port* and *Application exposed port*.
  - Click **Copy mappings** to create application mappings through this connection.

4. Now you have successfully exposed the application as you need. In our example, IBM Cloud Pak for AIOps is exposed to the public network, so you can use the application mapping address to access IBM Cloud Pak for AIOps.

Creating a Custom Instana connection

1. Ensure that your cluster has the following prerequisites installed and configured:
   - An Instana server is installed and runs in an Red Hat OpenShift Container Platform cluster.
   - An IBM Cloud Pak for AIOps cluster is installed and runs in a public cloud with Secure Tunnel enabled.
2. Create Secure Tunnel Connection for Instana:
   1. From the IBM Cloud Pak Automation console main navigation menu, click **Administration** > **Secure Tunnel**.
   2. On the *Secure Tunnel* page, click **Create connection**.
   3. Click the **Create connection** link under *Custom* section and then click **Get started**.
   4. On the *Name and tags* page, input a name in **Connection name** field, for example `instana-via-tunnel`, click **Next**.
   5. Choose **Connector in private network initializes connection to Secure Tunnel in public network`** in *Direction*, click **Next**.
   6. On *Connector settings* page, choose **Host machine**. Input the hostname or IP address of the host machine where you installed Secure Tunnel connector. Click **Add** > **Next**.
   7. On *Secure Tunnel settings* page, click **Next**.
   8. On *Allowed list* page, input the IP address and port of Instana server, for example `9.46.108.75` and `443`. Click **Add** button and click **Create connection**.
   9. Wait for the *Next steps* page display, then click **Close**.
   10. Download the Secure Tunnel Connector Installation package by using one of the following options:
       - If you are still on the *Next steps* page, you can download the installation package by clicking download link under `Install a connector`
       - If you are not in the *Next steps* page yet, go to **Manage Connection** page, find the `to-instana-through-ocp` row, and click the **three dots** in the **Actions** column, then click **Install connector** link. You can now download the installation package by clicking the download link in the open panel.
3. Install Secure Tunnel Connector to a host machine:
   1. refer to Install the Secure Tunnel Connector to a host machine (virtual machine or physical machine)
   2. On the *Secure Tunnel* page, the *Status* column of `instana-via-tunnel` connection is green.
4. Create Application Mapping to expose Instana
   1. From the *Secure Tunnel* page, click the **Application mapping** column on the row `instana-via-tunnel`.
   2. On the *Application mappings for instana-via-tunnel* page, click **Create** button.
   3. On the *Create application mapping* page, click **Next**.
   4. On the *Name and tags* page, input a name, for example `instana` then click **Next**.
   5. On *Direction* page, choose **From connector side to Secure Tunnel side** and click **Next**.
   6. On *Application details* page, input Instana server IP address and port, for example `9.46.108.75` and `443` and click **Add** button.

7. On *Access rules (optional)* page click **Create**.
8. On the *Application mappings for instana-via-tunnel* page, wait for a few minutes until the `Status` column of `instana` row become `Ready`.
9. In the *Application mapping address* column, **click the information button(a circle with an `i` in it)**, and copy the address in the open message. For example, `sre-tunnel-c60f3456bad94a10-svc.cp4waiops:443`. Save it to a note pad.

5. Create an Instana connection:

    1. From the IBM Cloud Pak Automation console navigation menu, click **Define** > **Data and tool connections**.
    2. On the Data and tool connections, click **Add connections** button.
    3. On the *Add connections* page, click **Add connection** link under *Instana* icon, then click **Connect** button.
    4. On the *Instana* page, in **Add connections**, provide the following fields:
        - input Name, for example `tunnel-2-instana-public`
        - input Description, for example `Connect to an Instant in public cloud through the Secure Tunnel`
        - input Endpoint, by using the `Application mapping address` copied from Secure Tunnel Application Mapping, for example `https://sre-tunnel-c60f3456bad94a10-svc.cp4waiops:443`.
        - input API Token, by using the API Token find from your Instana server, for example `W6p9WimUT5aYQ_skNu0W9A`
        - Click **Next**
    5. On the *Collect topology data* page, click **Next**.
    6. On the *Collect event data* page, click **Next**.
    7. On the *Collect metric data* page, click **Done**.
    8. In the Data and tool connections, a message indicates that connection is completed.
    9. Click **Instana** link, in the *Instana**page to see the `tunnel-2-instana-public` connection.

6. Verify Instana Connection

    1. From the IBM Cloud Pak Automation console navigation menu, click **Operate** > **AI model management**.
    2. On the AI Model Management page, click **Data assets** tab.
    3. The Data flow status column for the `tunnel-2-instana-public` connection should display a status of Running.

Viewing Secure Tunnel

On the Secure Tunnel web console, you can get the application mapping URL that can be accessed by your integration. In addition, you can view the configuration details and status of all tunnel connections and application mappings. If you have not accessed the Secure Tunnel web console previously, read this.

- Secure Tunnel Components
- Viewing tunnel connection list
- Viewing the application mappings
- Certificate expiration

Secure Tunnel Components

There are four components on the Secure Tunnel web console that you can use to manage your connections. To access these, go to **Administration** > **Secure Tunnel**.

- **Manage connection**

  Use **Manage connection** to manage your connections to remote networks, A connection links the network containing the Secure Tunnel to a remote network. Your applications can reside on either side of the connection. Secure Tunnel includes connection templates for common scenarios(Chatops, Vmware, Turbonomic). You can also create custom connections to support additional scenarios.

- **Manage template**

  Use **Manage template** to manage your templates. This tool can help you create a new connection quickly. Secure Tunnel has pre-defined templates for common scenarios, but users can also create templates for their special

requirement. Find out more about using templates

- **Audit operations**

  Use **Audit operations** to check Secure Tunnel configuration changes, such as tunnel connection or application mapping configuration changes for auditing. Find out more about auditing Secure Tunnel

- **Audit traffic**

  Use **Audit traffic** to check open/close connection events traffic and see the total amount of data that is sent or recieved from the specific connection listed in the close event. Find out more about auditing Secure Tunnel

Viewing tunnel connection list

Go to **Administration** > **Secure Tunnel**. In the *Manage connections* page, you can see the tunnel connection list with the following columns:

- **Name**: Name of the tunnel connection. Click the name of the tunnel connection in the *Manage connections* page and you are redirected to the *Application mappings* page and the *Connection details* page.
- **Connectors**: The number of connections. This indicates how many Connectors connect to the tunnel worker.
- **Application mapping(valid/total)**: The number of valid application mappings and the total number of application mappings that are passing this connection. You can click the entry in this column to go to the application mappings page for more details. See Viewing the application mappings.
- **Direction**: The relationship between the Secure Tunnel and the Connector. A Connector in a private network initializes a connection to a Secure Tunnel in a public network or a Secure Tunnel in a private network initializes a connection to a Connector in a public network
- **Allowed list**: The IP addresses and ports that are exposed across this connection by using application mappings.
- **Created**: Date and time Connector was created.
- **Certificate expiration**: Expiration time of certificates.
- **Status**: Indicates whether the tunnel connection is accessible. Five kinds of status are provided:
  - `Pending`: This means that the tunnel connection is creating the tunnel worker. When the connection in in `Pending` status, you cannot install the Connector for the connection.
  - `Waiting`: This means that tunnel worker is ready, but no Connector is connecting to it. You need to install a Connector and create application mapping for this connection.
  - `Ready`: This means that the connection is ready.
  - `Error`: This means that there are some errors with the tunnel connection. Hover over the status to see the detailed error message.
  - `Incomplete`: This means that the application mapping for the connection is invalid.
- **Tags**: Used to signify the tunnel connection.

Viewing the application mappings

Go to **Administration** > **Secure Tunnel**. This opens the *Manage connections* page. Click the number in the **Application mappings (valid/total)** column. You are redirected to the *Application mappings* page. This provides details of all the application mappings of a connection.

- **Name**: Name of the application mapping.
- **Creator**: User that creates the application mapping.
- **Application original address**: The private address of the application that you want to expose.
- **Application mapping address**: The application mapping URL. You can access the private application by this mapping address.
- **Network policy**: To control the traffic flow at the IP address or port level for this application.
- **Created**: Time when the application mapping was created.
- **Status**: Indicates whether the application mapping is accessible. Three kinds of status are provided:

    - `Error`: This means that there are some errors with the application mapping. You can move the mouse up to see the detailed error message.
    - `Incomplete`: If the application mapping status is `Incomplete`, hover your mouse pointer over the *Status* column to discover the problem. It could be that you did not install the Connector or that Connector is still waiting to connect. If a port conflict is indicated, uninstall the Tunnel connector, resolve the port conflict, and then re-install the Tunnel connector.
    - `Ready`: This means that the application mapping address is accessible.
- **Tags**: Tags of the application mapping.

Certificate expiration

When the certificate for the Secure Tunnel connection expires, the IBM Certificate Manager renews it. The Secure Tunnel worker server automatically reloads the certificate, but the Secure Tunnel connector does not. You must download a new Secure Tunnel connector installation script, and then uninstall and reinstall the Secure Tunnel connector. For more information, see the instructions in Install the Secure Tunnel Connector.

Modifying Secure Tunnel

In the Secure Tunnel web console you can modify the configurations for a tunnel connection and application mapping to meet your needs. (Read this if you have not used Secure Tunnel web console previously).

- Configuring connections
- Managing Template
- View Template
- Delete Template
- How to obtain the application mapping address

Configuring connections

- **Configuring tunnel connections**
    - **Edit connection**: To modify the connection, locate the connection, and click **Action** > **Edit** connection.
    - **Renew certification**: To renew the certificate of the tunnel connection, locate the connection, and click **Action > Renew** certificate. Then, the certificate of the tunnel connection is updated, and the certificate expiration time field is changed.
    - **Delete connection**: To delete a tunnel connection, locate the connection, and click **Action > Delete connection**.
- **Configuring application mapping**
    - **Edit application mapping**: To edit the application mapping, locate the application mapping, and click **Action > Edit**.
    - **Delete application mapping**:To delete the application mapping, locate the application mapping, and click **Action > Delete**.
- **Installing Secure Tunnel connector**
    - Follow this guide

Managing Connection Templates

A template includes pre-defined connection settings and one or more application mappings. Examples of pre-defined templates available in IBM Cloud Pak for AIOps are ChatOps and Vmware, which you can access in **Manage Connections** > **Create connection**.

You can use a template to quickly create a new connection, or add application mappings to an existing connection. You can easily modify the application mappings to meet the unique requirements of the connection.

Create Template

Click **Create template** button to open the wizard page:

1. Provide a name and a description for this template.
2. Select the connection upon which the template is based.
3. Select the application mappings for the template.

Once you create a template, you can use it to simplify the creation of future connections.

View Template

Click **Action** in the template column. Then, click **View details** to view the template details.

Delete Template

Click **Action** on the template column. Then, click **Delete** to delete the template.

How to obtain the Application URL mapping address

1. From the Secure Tunnel console, select **Manage connections** tab.
2. Click the **Application mappings** item of the tunnel connection.
3. Choose the **Application mapping** by name from the **Application mapping** list page.
4. Copy the address from the **Application mapping address** item.

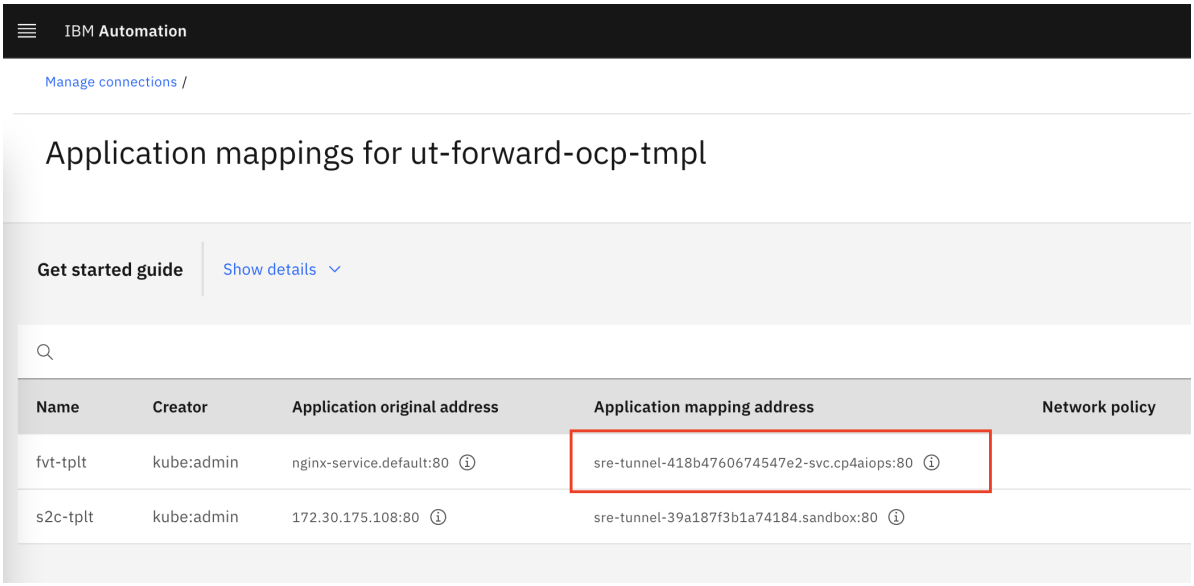


Figure. Application mapping

Auditing Secure Tunnel

You can check Secure Tunnel configuration changes, such as tunnel connection or application mapping configuration changes for auditing.

How to audit Secure Tunnel

Go to **Administration** > **Secure Tunnel** on IBM Cloud Pak for AIOps console. You see the **Audit operations** and **Audit traffic** tabs.

1. In the *Audit operations* tab, you can see the operation logs, such as deleting a connection or creating a connection. You can filter the logs by changing the start time and end time in the **Start time** and **End time** fields. **Note:** You can click the **Operation details** column to see more about the tunnel operation.

2. In the *Audit traffic* tab, the tunnel **`open/close`** connection events are listed, and a total **`send/receive`** data amount for the specific connection is listed in the **`close`** event. You can filter the logs by changing the start time and end time in

the **Start time** and **End time** fields.

Configuring Secure Tunnel audit logs

Secure Tunnel can send audit logs to its container log, a MongoDB database, and a remote syslog server. By default, Secure Tunnel sends audit logs to its container log, and thus you can't see audit logs on the Secure Tunnel console. To check audit logs on the Secure Tunnel console, you need to send audit logs to MongoDB. To check audit logs on the query user interface of a specific syslog, you need to send audit logs to the remote syslog server.

Configuring with the operator UI when creating a Secure Tunnel installation

1. When you create the Secure Tunnel instance from your OpenShift console, click **Operators** > **Installed Operators**.
2. From the **Project** menu, select the project that you want to create the Secure Tunnel instance in. Use the project that you created earlier in **Create a Secure Tunnel project (namespace)**.
3. Select **IBM Secure Tunnel** operator.
4. Click **Create Installation** on the *Tunnel* box. The default **Form View** is displayed. Edit the **Audit logs** to configure the audit log receivers of the Secure Tunnel.
5. *Show audit logs in the container log:* If the Secure Tunnel should send audit logs to its container log(the pod terminal).
6. *Audit Receiver:* to add a **MongoDB audit log receiver** or a **Syslog audit log receiver**:
    1. Click the **MongoDB receiver** > **Add MongoDB receiver** button. To add a **MongoDB audit log receiver** > click **MongoDB Information**. Enter the following values:
        - **Receiver name:** The name of the MongoDB receiver, it's the unique name of the audit log receiver. for example: **`mongodb-receiver-1`**.
        - **TLS Secret:** if you want the MongoDB to support TLS, you should create a TLS Kubernetes secret for the MongoDB TLS certification, and enter the *k8s secret name* to here. The Kubernetes secret should be in the Secure Tunnel namespace and should contain the "ca.crt", "tls.crt" and "tls.key" of the mongoDB TLS. TLS secret example:

```
kind: Secret
apiVersion: v1
metadata:
name: mongodb-tls-certificate
namespace: tunnel-stand-alone
data:
ca.crt:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURDVENDQWZHZ0F3SUJBZ0lRTjRtVDZOTFgxZjF
1K0pSVFR4LzRoakFOQmdrcWhraUc5dzBCQVFzRkFEQWcKTVI0d0hBWURWUVFERXhWMGRXNXVaV3d0Y2
1WaFpHbHVaWE56TFdObGNuUXdIaGNOTWpJd01qRXlNVFV4TURReQpXaGNOTXpJd01qRXdNVFV4TURRe
VdqQWdNUjR3SEFZRFZRUURFeFYwZFc1dVpXd3RjbVZoWkdsdVpYTnpMV05sCmNuUXdnZ0VpTUEwR0NT
cUdTSWIzRFFFQkFRVUFBNElCRHdBd2dnRUtBb0lCQVFEUFA0MHQ4WG5aWmZNTUdzVWoKZ0FlNkpJUE9
0OXlWVVV1Y2ppM3dHZVJwaUszVDdBY2EvQzdTUHprenE4YkRyZnVwMUZDc0hYTFo0RmFSeS9BVQpTV0
RtQ3NPZXpNYzFjNkYyUkJ0Y1djOGF4VDRlVEI2VE9DeTQ3cFdaOVJNVjJnQUNJQktDc1JWZHIzZ1JQY
1F6CmtCbWptSERFRTVtOUlrNVFSYktfMWl2SDV0NjhXMHFaNEZWUDNuQS9WSXhBaXlHNDczYVFFbHg5
NFc1OFV4a24KRkNOSzdEZjkycE1GUS95QmdlMEkvb1VUWUpicHdQMDBMTE1XTTgyalJ1OFJ6Q0F6SUh
aNXVTUXp0UGhYQWZQVQpwS3NtRVRNVms1a2MvQnN1VW9QWDhGeFl1di85RHNiOEw1K21sSFJZcFRNUV
hhWWs0cG1nODdYSFF5RDdnb1V1CnJmejdBZ01CQUFHalB6QTlNQTRHQTFVZER3RUIvd1FFQXdJQ3BEQ
VBCZ05WSFJNQkFmOEVCVEFEQVFIL01Cb0cKQTFVZEVRUVRNQkdDQ1d4dlkyRnNhRzl6ZEljRWZ3QUFB
VEFOQmdrcWhraUc5dzBCQVFzRkFBT0NBUUVBQlBxbwpuL01NODV6QnBBZ2poUTVIYkdOU0l2MVBOWkw
0MmJ3blhkbVRFQVR3WHFQbzlUWmIvd0htS3hUUFJ4TjhDLzRvCmFxaEdkMmRPb21zTjJ1SStNTE5JMj
ZVak1VNWFxVS8zajdMYzV1QmJQVHRXVURnZXE2aTFUdXVDNncrd1hTRjYKRlV4YTFVYjF2dWZkZzl3O
HREWlF2dWlnQlkwaCtYWnNjckFQVVdzNGZQMnpTQytYSElSSCtENXltR1NlWk4zeQpoam9LdGVCQTFV
NlpZemtWVDVHV1QwK1p0MVBXUS8wQmQ4RUxzSHRjOHkrVGZVckZvaUkxNlRWYm8vUDlJNlhVCnBhaXU
2SFJFUFh6MFdISnppSUZyRUE2dFZIdmJXMzdFMnJmR0ZSUWxCSnM1T0RyWTJHQndKM0RCd3MzbGN4cW
UKZURyQTNzaWRZWE9TSThvMFRRPT0KLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo=
tls.crt:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURDVENDQWZHZ0F3SUJBZ0lRTjRtVDZOTFgxZjF
1K0pSVFR4LzRoakFOQmdrcWhraUc5dzBCQVFzRkFEQWcKTVI0d0hBWURWUVFERXhWMGRXNXVaV3d0Y2
1WaFpHbHVaWE56TFdObGNuUXdIaGNOTWpJd01qRXlNVFV4TURReQpXaGNOTXpJd01qRXdNVFV4TURRe
VdqQWdNUjR3SEFZRFZRUURFeFYwZFc1dVpXd3RjbVZoWkdsdVpYTnpMV05sCmNuUXdnZ0VpTUEwR0NT
cUdTSWIzRFFFQkFRVUFBNElCRHdBd2dnRUtBb0lCQVFEUFA0MHQ4WG5aWmZNTUdzVWoKZ0FlNkpJUE9
```

```
0OXlWVVV1Y2ppM3dHZVJwaUszVDdBY2EvQzdTUHprenE4YkRyZnVwMUZDc0hYTFo0RmFSeS9BVQpTV0
RtQ3NPZXpNYzFjNkYyUkJ0Y1djOGF4VDRlVEI2VE9DeTQ3cFdaOVJNVjJnQUNJQktDc1JWZHIzZ1JQY
1F6CmtCbWptSERFRTVtOUlrNVFSYktFMWl2SDV0NjhXMHFaNEZWUDNuQS9WSXhBaXlHNDczYVFFbHg5
NFc1OFV4a24KRkNOSzdEZjkycE1GUS95QmdlMEkvb1VUWUpicHdQMDBMTE1XTTgyalJ1OFJ6Q0F6SUh
aNXVTUXp0UGhYQWZQVQpwS3NtRVRNVms1a2MvQnN1VW9QWDhGeFl1di85RHNiOEw1K21sSFJZcFRNUV
hhWWs0cG1nODdYSFF5RDdnb1V1CnJmejdBZ01CQUFHalB6QTlNQTRHQTFVZER3RUIvd1FFQXdJQ3BEQ
VBCZ05WSFJNQkFmOEVCVEFEQVFIL01Cb0cKQTFVZEVRUVRNQkdDQ1d4dlkyRnNhRzl6ZEljRWZ3QUFB
VEFOQmdrcWhraUc5dzBCQVFzRkFBT0NBUUVBQlBxbwpuL01NODV6QnBBZ2poUTVIYkdOU0l2MVBOWkw
0MmJ3blhkbVRFQVR3WHFQbzlUWmIvd0htS3hUUFJ4TjhDLzRvCmFxaEdkMmRPb21zTjJ1SStNTE5JMj
ZVak1VNWFxVS8zajdMYzV1QmJQVHRXVURnZXE2aTFUdXVDNncrd1hTRjYKRlV4YTFVYjF2dWZkZzl3O
HREWlF2dWlnQlkwaCtYWnNjckFQVVdzNGZQMnpTQytYSElSSCtENXltR1NlWk4zeQpoam9LdGVCQTFV
NlpZemtWVDVHV1QwK1p0MVBXUS8wQmQ4RUxzSHRjOHkrVGZVckZvaUkxNlRWYm8vUDlJNlhVCnBhaXU
2SFJFUFh6MFdISnppSUZyRUE2dFZIdmJXMzdFMnJmR0ZSUWxCSnM1T0RyWTJHQndKM0RCd3MzbGN4cW
UKZURyQTNzaWRZWE9TSThvMFRRPT0KLS0tLS1FTkQgQ0VSVElGSUNBVEUtLS0tLQo=
tls.key:
LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktLS0tLQpNSUlFb3dJQkFBS0NBUUVBenorTkxmRjUyV1h
6REJyRkk0QUh1aVNEenJmY2xWRkxuSTR0OEJua2FZaXQwK3dICkd2d3Uwajg1TTZ2R3c2MzdxZFJRck
IxeTJlQldrY3Z3RkVsZzVnckRuc3pITlhPaGRrUWJYRm5QR3NVK0hrd2UKa3pnc3VPNlZtZlVURmRvQ
UFpQVNnckVWWGE5NEVUM0VNNUFabzVod3hCT1p2U0pPVUVXeWhOWXJ4K2J1dkZ0SwptZUJWVDk1d1Ax
U01RSXNodU85MmtCSmNmZUZ1ZkZNWkp4UWpTdXczL2RxVEJVUDhnWUh0Q1A2RkUyQ1c2Y0Q5Ck5DeXp
GalBObzBidkVjd2dNeUIyZWJra003VDRWd0h6MUtTckpoRXpGWk9aSFB3YkxsS0QxL0JjV0xyLy9RN0
cKL0MrZnBwUjBXS1V6RUYybUpPS1pvUE8xeDBNZys0S0ZMcTM4K3dJREFRQUJBb0lCQVFDYmZ6OWFQV
mdET1c4RwpvZjlRQjN2RXVpajdMUkNnS0FMblQvWS9zKzJoRVIvN1paaGZNdCtlZURsZS8zOSt5a2VN
d1BSV2J6SUJMMFZFCjF4TW01WGVDb29qcjJSR09sVmVMK01YZW9MTzNqazV4UlFPYnlnWWtLejZyNlN
tTXQ2aFI3K1p6a0pKTFhLQmwKcjBaS3VjL0M0MW1ZaEd4UmNDLzZJcEV6QWM2UGlOcVZaVmJEdEl1Sz
dMNzBsbzRlQ09VSzl6WUNJcG9IcUc0MgprSjIyczIva2ppVkRiTmwyRVk0bVkrRm1RemgxMVEyaE5VZ
k5FS2YwZzk0b1pMdWdsWWVRLzhzV1h5UjExbWQ3CjgwbEF0eUYzWWJEaVNWOWFWMHNsUUhuZjYvSmFF
cWxySWxTa2M5OVdRN1ZpQUFyaEpSVlZjR29xQ0d0a3EyOFIKd3ZqenJrbWhBb0dCQVB0a09VblhST09
3VkRWdWp0aUl6RDFnZG9iSVptclNjZm5CZXlRUlFmRExJRXlZcVI3OQpzZ3VsMTVER1lBQXJySnR5NX
RiUGdUZ0JLclUxeS93aDExd2I2NW5wT3FaVmp6MDZzMXA5eFpIb3o0RVBYNmZZCkEranN3ZmRVMDk1b
3BtMnpxSy9DME01WlhvTEVEcTRsV0xVL21kb203WWo2T21FdzlXUm1zTVBuQW9HQkFOTU0KS2dVTGIz
SGdPSERVQ0s1M2daRkwrZVpoaGNWdENBSllHU05CN0JHdDVHWnJuZmVMVURJVThhL0hEK1YyNkdGago
0dWwzNDRrM1ZjSTJITnkxMlViQTVBb3poVFFQTWlLNUNwUW5hTjFRSmk4ejNDMlBFa2c1TjBHbEM5eW
poZHgvClJpV0VLNncrT01jV3ZHVWlGSmFncUR0a2hQaGJ0aDRVZUFvRzBsdk5Bb0dBRXhWZG4vUmVmM
lpRVGlvUlgvQ3EKaUhJOUxTZ2pWaktaZ2xza3JPMCtjemdJUUtoS0ZsY3QzK0o4QmJwNWFTbm5sWGxZ
ZXJBUDRCOStqWGtDL2lYWApnbCtBT0JQVFp2TVJ4NVVnTGx0ajF4cTBQZGpiNnhMajRSZXY5Y1dUdTh
0Mzd0WDNsVGdWdjhSYS8xUkF4UW1DCnZXZVJsMVRqTFVYSmpyMDNzNkowQXJzQ2dZQmkybHJBUENmNk
5lVVo3Q2NZMlQ4bU9Ha1lJT3ZRbW81YTByc0wKaWJBeVk3b013ekdkcG5KczB1U1Z5ZTdITTFnTGJnY
jJCa0xjbmIrY2xVNFZqMVB6VVJ3L0x2eWdjdEd2OCs1aAprTWdNaWFNcER1aDlxbnNNdTBnSGhuQTA2
NThpdGJhOGtka0c5Nm9PanJhMXhDM1NrL1BZeGUrR2lTYVhsa3pYCm5YUmdyUUtCZ0VmSkZEM0p1N1l
HT3doUkRER2dOMUdRWkd0TFFzbFdUUnBYWlRtWDFiT016VWxSeUIzaUdvdHoKalJHbWdYVE9lQzJIK3
dBczQzdWJHbklrbThrVk1OaWtXSlI0a3VOeDViOEFQUkc1R2l1MysvKzJHU0NuZ1NGUAp6TVU1dWpEN
2lNQklBL0VGMDhMSzVJY2I5bTJJYVo0ZmJqNmdYZU5oeTdNdHh2VDR0eGdDCi0tLS0tRU5EIFJTQSBQ
UklWQVRFIEtFWS0tLS0tCg==
type: kubernetes.io/tls
```

- **Database name:** The database name of the MongoDB receiver, for example: **`mongodb-audit-log-receiver-1`**
- **Adress:** The MongoDB receiver address. For example: **`<IP or hostname>:27017`**
- **MongoDB log in Username, Password Secret:**
  - By default, MongoDB does not require a username and password to access, but it can be enabled by configuring the authentication. If the MongoDB has a username and password, you should create a Kubernetes secret that contains **`user: <MongoDB username>`** and **`password: <MongoDB password>`** in the Secure Tunnel namespace, then enter the secret name here.

    Kubernetes secret example:

    ```
    kind: Secret
    apiVersion: v1
    metadata:
      name: mongodb-credential
      namespace: tunnel-stand-alone
    data:
      password: YmJiYg==
      user: YWFhYQ==
    type: Opaque
    ```

2. Click **Syslog receiver** > **Add Syslog receiver** to add a **Syslog audit log receiver** and enter the following values:

- *Receiver name:* the name of the Syslog receiver. It's a unique name of the audit log receiver. for example: `syslog-receiver-1`.
- *Address:* The syslog server address. for example: `<IP or hostname>:5514`

Configuring YAML to update the existing Tunnel instance

1. From the Openshift console, click **Operators** > **Installed Operators**.
   - From the **Project** dropdown menu, select the project that you created the Secure Tunnel instance in.
   - Click the **IBM Secure Tunnel** > **Tunnel** tab > click the existing **sre-tunnel** instance > change to **YAML** tab.
2. For Secure Tunnel to send audit logs to its container log, configure the **YAML** file:

```
spec:
  ...
  showInContainerLog: true
  ...
```

3. For Secure Tunnel to send audit logs to MongoDB, configure the **YAML** file:

```
spec:
  ...
  auditReceiver:
    mongoDB:
      - mongoAddress:
          mongoAddress: '<IP or hostname that the MongoDB running in>:27017'
          mongoDatabaseName: secure-tunnel-001
          name: mongodb-audit-log
          passwordSecretName: <The name of the Kubernetes secret that saved the MongoDB
username and password>
          tlsSecret: <The name of the Kubernetes secret that saved the SSL certificate of
the MongoDB service>
  ...
```

4. For Secure Tunnel to send audit logs to a remote syslog server, configure the **YAML** file:

```
spec:
  ...
  auditReceiver:
    syslog:
      - name: syslog
        syslogAddress: '<IP or hostname that the Syslog service running in>:5514'
  ...
```

Tip: It's easier to configure the audit log with the Configuring with the operator UI when create Secure Tunnel Installation method, so you can configure it from there.Switch to the **YAML view**, obtain the configuration yaml and update the existing Secure Tunnel instance yaml to configure the audit log.

Restarting the Cloud Pak for AIOps cluster

Before a known maintenance window or outage, you might need to shut down the cluster where IBM Cloud Pak for AIOps is installed, and then restart it.

Warning: If you need to shut down the cluster where IBM Cloud Pak for AIOps is installed, then you must shut down the stateful services using the procedure below. Failure to do so may result in data loss or corruption.

Procedure

1. Set up SSH connections to the cluster
2. Check the certificates
3. Scale down the StatefulSets and drain the nodes
4. Shut down the cluster
5. Restart the cluster
6. Scale up the StatefulSets

1. Set up SSH connections to the cluster

1. To allow SSH connection to each of the hosts in the cluster, you need to set up SSH connections to the cluster by adding some configuration to the **~/.ssh/config** file on the host that you are accessing the cluster from.

   Edit the **~/.ssh/config** file by adding the following configurations for each master, worker, and api node in your cluster.

   Example:

   ```
   Host acmecluster-api
   HostName api.acmecluster.cp.acme.com
   User root
   Port 22

   Host acmecluster-master0
   HostName master0.acmecluster.cp.acme.com
   User core
   Port 22
   ProxyJump acmecluster-api
   IdentityFile ~/.ssh/id_rsa_os

   Host acmecluster-worker0
   HostName worker0.acmecluster.cp.acme.com
   User core
   Port 22
   ProxyJump acmecluster-api
   IdentityFile ~/.ssh/id_rsa_os
   ```

2. To allow connections to the worker and master nodes, copy the **id_rsa** file from the **cluster-api** server to the **/tmp** directory on the host that you are accessing the cluster from, by running the following command:

   ```
   scp root@<clustername>-api:/root/.ssh/id_rsa  /tmp/id_rsa
   ```

3. Rename the file to **id_rsa_os**, and copy it to the **~/.ssh/** directory. You can access each of the nodes in the **~/.ssh/config** file by using SSH now.

   You can test the SSH connection by running the following command:

   ```
   ssh core@<clustername>-worker0
   ```

   Note: Before you continue to the next step, test the SSH connections to all the worker, master, and api nodes.

2. Check the certificates

Ensure that none of the certificates have issues or are expired by running the following command:

```
echo -e "NAMESPACE\texNAME\tEXPIRY" && oc get secrets -A -o go-template='{{range .items}}{{if
eq .type "kubernetes.io/tls"}}{{.metadata.namespace}}{{" "}}{{.metadata.name}}{{" "}}{{index
.data "tls.crt"}}{{"\n"}}{{end}}{{end}}' | while read namespace name cert; do echo -en
"$namespace\t$name\t"; echo $cert | base64 -d | openssl x509 -noout -enddate; done | column -t
```

Example output:

```
   NAMESPACE     NAME     EXPIRY
   cp4aiops      aimanager-aio-similar-incidents-service-cert    notAfter=Apr 25 10:15:07 2024
GMT
   cp4aiops      aiops-ai-model-ui-tls-secret    notAfter=Apr 25 09:44:59 2024 GMT
   cp4aiops      aiops-akora-ui-tls-secret       notAfter=Apr 25 09:44:59 2024 GMT
   cp4aiops      aiops-application-details-ui-tls-secret         notAfter=Apr 25 09:44:59 2024
GMT
   cp4aiops      aiops-base-ui-tls-secret        notAfter=Apr 25 09:45:16 2024 GMT
   cp4aiops      aiops-connections-ui-tls-secret         notAfter=Apr 25 09:44:59 2024 GMT
   cp4aiops      aiops-ir-analytics-classifier-tls       notAfter=Apr 25 10:12:12 2024 GMT
   cp4aiops      aiops-ir-analytics-probablecause-tls    notAfter=Apr 25 10:12:15 2024 GMT
   cp4aiops      aiops-ir-core-archiving-tls     notAfter=Apr 25 10:03:21 2024 GMT
   cp4aiops      aiops-ir-core-cem-users-tls     notAfter=Apr 25 10:03:20 2024 GMT
```

```
   cp4aiops     aiops-ir-core-esarchiving-tls   notAfter=Apr 25 10:03:21 2024 GMT
   cp4aiops     aiops-ir-core-ncobackup-tls     notAfter=Apr 25 10:02:08 2024 GMT
   cp4aiops     aiops-ir-core-ncodl-api-tls     notAfter=Apr 25 10:03:20 2024 GMT
   cp4aiops     aiops-ir-core-ncodl-std-tls     notAfter=Apr 25 10:03:21 2024 GMT
   cp4aiops     aiops-ir-core-ncoprimary-tls    notAfter=Apr 25 10:00:24 2024 GMT
   cp4aiops     aiops-ir-core-rba-as-tls        notAfter=Apr 25 10:03:21 2024 GMT
   cp4aiops     aiops-ir-core-rba-rbs-tls       notAfter=Apr 25 10:03:21 2024 GMT
   cp4aiops     aiops-ir-lifecycle-eventprocessor-ep-client-cert-kp     notAfter=Jul 25
09:57:52 2022 GMT
   cp4aiops     aiops-ir-lifecycle-eventprocessor-ep-internal-cert-kp   notAfter=Jul 25
09:57:52 2022 GMT
   cp4aiops     aiops-ir-lifecycle-eventprocessor-ep-ss-cacert-kp       notAfter=Jul 25
09:57:34 2022 GMT
   cp4aiops     aiops-ir-lifecycle-policy-grpc-clt-svc  notAfter=Jul 25 09:59:01 2022 GMT
   cp4aiops     aiops-ir-lifecycle-policy-grpc-svc      notAfter=Apr 25 09:58:54 2024 GMT
   cp4aiops     aiops-ir-ui-api-graphql-tls-secret      notAfter=Apr 25 09:45:00 2024 GMT
   cp4aiops     aiops-topology-alm-observer-cert        notAfter=Apr 25 10:38:36 2024 GMT
   cp4aiops     aiops-topology-ansibleawx-observer-cert         notAfter=Apr 25 10:38:36 2024
GMT
   cp4aiops     aiops-topology-appdynamics-observer-cert        notAfter=Apr 25 10:38:31 2024
GMT
   cp4aiops     aiops-topology-aws-observer-cert        notAfter=Apr 25 10:38:32 2024 GMT
   cp4aiops     aiops-topology-azure-observer-cert      notAfter=Apr 25 10:38:34 2024 GMT
   cp4aiops     aiops-topology-cienablueplanet-observer-cert    notAfter=Apr 25 10:38:32 2024
GMT
   cp4aiops     aiops-topology-ciscoaci-observer-cert   notAfter=Apr 25 10:38:32 2024 GMT
   cp4aiops     aiops-topology-contrail-observer-cert   notAfter=Apr 25 10:38:34 2024 GMT
   cp4aiops     aiops-topology-dns-observer-cert        notAfter=Apr 25 10:38:32 2024 GMT
   cp4aiops     aiops-topology-docker-observer-cert     notAfter=Apr 25 10:38:31 2024 GMT
   cp4aiops     aiops-topology-dynatrace-observer-cert  notAfter=Apr 25 10:38:34 2024 GMT
   cp4aiops     aiops-topology-file-observer-cert       notAfter=Apr 25 09:56:57 2024 GMT
   cp4aiops     aiops-topology-googlecloud-observer-cert        notAfter=Apr 25 10:38:35 2024
GMT
   cp4aiops     aiops-topology-ibmcloud-observer-cert   notAfter=Apr 25 10:38:32 2024 GMT
   cp4aiops     aiops-topology-itnm-observer-cert       notAfter=Apr 25 10:38:31 2024 GMT
   cp4aiops     aiops-topology-jenkins-observer-cert    notAfter=Apr 25 10:38:31 2024 GMT
   cp4aiops     aiops-topology-junipercso-observer-cert         notAfter=Apr 25 10:38:32 2024
GMT
   cp4aiops     aiops-topology-kubernetes-observer-cert         notAfter=Apr 25 09:56:56 2024
GMT
   cp4aiops     aiops-topology-layout-cert      notAfter=Apr 25 09:56:56 2024 GMT
   cp4aiops     aiops-topology-merge-cert       notAfter=Apr 25 09:56:56 2024 GMT
   ibm-common-services  auth-pdp-secret         notAfter=Jul 25 09:37:07 2022 GMT
   ibm-common-services  common-web-ui-cert      notAfter=Jul 25 09:37:13 2022 GMT
   ibm-common-services  cs-ca-certificate-secret        notAfter=Apr 25 09:33:51 2024 GMT
   ibm-common-services  cs-webhook-cert         notAfter=Apr 25 09:29:22 2024 GMT
   ibm-common-services  iam-pap-secret  notAfter=Jul 25 09:37:08 2022 GMT
   ibm-common-services  ibm-iam-bindinfo-platform-auth-secret   notAfter=Jul 25 09:37:20 2022
GMT
   ibm-common-services  ibm-license-service-cert        notAfter=Apr 25 09:32:11 2024 GMT
   ibm-common-services  icp-management-ingress-tls-secret       notAfter=Apr 26 09:36:56 2023
GMT
   ibm-common-services  icp-mongodb-client-cert         notAfter=Apr 25 09:37:41 2024 GMT
   ibm-common-services  identity-provider-secret        notAfter=Jul 25 09:37:22 2022 GMT
   ibm-common-services  mongodb-root-ca-cert    notAfter=Apr 25 09:35:54 2024 GMT
   ibm-common-services  platform-api-secret     notAfter=Jul 25 09:37:08 2022 GMT
   ibm-common-services  platform-auth-secret    notAfter=Jul 25 09:37:20 2022 GMT
   ibm-common-services  platform-identity-management    notAfter=Jul 25 09:36:46 2022 GMT
   ibm-common-services  route-tls-secret        notAfter=Apr 26 09:37:21 2023 GMT
```

If any certificates have issues or are expired, you can refer to Renew or re-create certificates in Openshift 4.x.

3. Scale down the StatefulSets and drain the nodes

1. Cordon the worker nodes by running the following commands:

```
for i in `seq 0 <number_of_worker_nodes>`
do
oc adm cordon worker$i.<clustername>
echo $i
```

```
sleep 2
done
```

Where **<number_of_worker_nodes>** is the number of worker nodes in your cluster.

2. Make a note of the number of replicas for each StatefulSet.

   Run the following command to see the number of replicas for StatefulSets in the IBM Cloud Pak for AIOps installation namespace.

   ```
   oc get statefulsets -n <namespace>
   ```

   Where **<namespace>** is the namespace (project) that your IBM Cloud Pak for AIOps installation is deployed in.

   Example output:

   ```
   NAME READY AGE
   aimanager-aio-luigi-daemon 1/1 3d22h
   aimanager-ibm-minio 1/1 3d22h
   aiops-ir-core-ncobackup 1/1 3d22h
   aiops-ir-core-ncoprimary 1/1 3d22h
   aiops-ir-lifecycle-eventprocessor-ep-jobmanager 1/1 3d22h
   aiops-ir-lifecycle-eventprocessor-ep-taskmanager 1/1 3d22h
   aiops-topology-cassandra 1/1 3d22h
   c-example-couchdbcluster-m 1/1 3d22h
   c-example-redis-m 3/3 3d23h
   c-example-redis-s 3/3 3d23h
   cp4waiops-eventprocessor-eve-29ee-ep-jobmanager 1/1 3d22h
   cp4waiops-eventprocessor-eve-29ee-ep-taskmanager 2/2 3d22h
   iaf-system-elasticsearch-es-aiops 1/1 3d23h
   iaf-system-kafka 1/1 3d23h
   iaf-system-zookeeper 1/1 3d23h
   sre-tunnel-controller 1/1 3d23h
   zen-metastoredb 3/3 3d23h
   ```

   Run the following command to see the number of replicas for the **icp-mongodb** StatefulSet in the **ibm-common-services** namespace.

   ```
   oc get statefulsets -n ibm-common-services
   ```

3. Scale down each of the StatefulSets that you identified in the previous step to 0.

   You can use the Cloud Pak for AIOps console, or create a shell script to do this.

   Example shell script:

   ```
   #!/bin/bash

   oc scale statefulsets icp-mongodb --replicas=0 -n ibm-common-services
   sleep 2
   oc scale statefulsets iaf-system-zookeeper --replicas=0 -n <namespace>
   sleep 2
   oc scale statefulsets iaf-system-kafka --replicas=0 -n <namespace>
   sleep 2
   oc scale statefulsets c-example-couchdbcluster-m --replicas=0 -n <namespace>
   sleep 2
   oc scale statefulsets iaf-system-elasticsearch-es-aiops --replicas=0 -n <namespace>
   sleep 2
   oc scale statefulsets aimanager-ibm-minio --replicas=0 -n <namespace>
   sleep 2
   oc scale statefulsets aiops-topology-cassandra --replicas=0 -n <namespace>
   sleep 2
   oc scale statefulsets c-example-redis-m --replicas=0 -n <namespace>
   sleep 2
   oc scale statefulsets c-example-redis-s --replicas=0 -n <namespace>
   sleep 2
   oc scale statefulsets zen-metastoredb --replicas=0 -n <namespace>
   sleep 2
   ```

   Where **<namespace>** is the namespace (project) that your IBM Cloud Pak for AIOps installation is deployed in.

Run the following command to check that the number of replicas for each of the StatefulSets is now 0.

```
oc get statefulsets -n <namespace>
```

Where **<namespace>** is the namespace (project) that your IBM Cloud Pak for AIOps installation is deployed in.

Example output:

```
Statefulset     Number of pods
iaf-system-kafka         0
iaf-system-zookeeper     0
c-example-couchdbcluster-m       0
iaf-system-elasticsearch-es-aiops        0
aimanager-ibm-minio      0
aiops-topology-cassandra         0
c-example-redis-m        0
c-example-redis-s        0
zen-metastoredb          0
icp-mongodb      0
```

4. After the StatefulSets are scaled down, drain the nodes by running the **drain** command manually, or by running a loop as follows:

```
for i in `seq 0 <number_of_worker_nodes>`
do
echo $i
oc adm drain worker$i.<clustername> --ignore-daemonsets --force --grace-period=30 --
delete-local-data
sleep 10
done
```

Where **<number_of_worker_nodes>** is the number of worker nodes in your cluster.

Note: Some pods, such as rook pods, do not stop because this would break the error budget. If this problem occurs, run the commands in each node until only the rook pods are left, and then stop the command and restart the script at the next worker node. This action is necessary because otherwise the drain command will continuously retry, and the command will never proceed to the next worker node.

4. Shut down the cluster

1. Shut down all of the worker nodes by running the following command on each of the worker nodes in your cluster:

```
ssh core@<worker> 'sudo halt -f'
```

Where **<worker>** is a worker node in your cluster.

Note: Ensure that an Ansible inventory has been created before you run this command.

2. After all of the worker nodes are shut down, shut down the master nodes.
    1. Run the following command to check that all of the worker nodes are shut down.

    ```
    oc get nodes
    ```

    2. Run the following command on each of the master nodes in your cluster.

    ```
    ssh core@<master> 'sudo halt -f' &
    ```

    Where **<master>** is a master node in your cluster.
3. Shut down the api node.

5. Restart the cluster

1. You must restart the cluster in the following order:

1. Restart the api host.
2. Restart the master nodes. Check whether all master nodes are in **`ready`** status by running the following command:

   ```
   oc get nodes
   ```

3. Restart the worker nodes. Check whether all worker nodes are in **`ready`** status by running the following command:

   ```
   oc get nodes
   ```

2. After all of the nodes are up, uncordon the nodes by running either of the following commands:

   ```
   for i in `seq 0 <number_of_worker_nodes>`
   do
   oc adm uncordon worker$i.<clustername>
   echo $1
   sleep 2
   done
   ```

   Where **`<number_of_worker_nodes>`** is the number of worker nodes in your cluster.

   ```
   oc adm uncordon <node_name>
   ```

   Then, you can see that the worker nodes have a status of **`Ready`** on the Red Hat OpenShift console, and the pods start.

6. Scale up the StatefulSets

1. Now you can scale up the StatefulSets based on the number of replicas that you noted in 3. Scale down the StatefulSets and drain the nodes.

   You must rescale the StatefulSets in the following order:

   - `icp-mongodb`
   - `aiops-topology-cassandra`
   - `iaf-system-zookeeper`
   - `iaf-system-kafka`
   - `c-example-couchdbcluster-m`
   - `iaf-system-elasticsearch-es-aiops`
   - `c-example-redis-m`
   - `c-example-redis-s`

   Then, rescale the rest of the StatefulSets.

   - `aimanager-ibm-minio`
   - `zen-metastoredb`

   Run the following command to rescale each StatefulSet:

   ```
   oc scale statefulsets <statefulset> --replicas=<number of replicas> -n <namespace>
   ```

   Where:

   - **`<statefulset>`** is the StatefulSet to be scaled up
   - **`<number_of_replicas>`** is the number of replicas the StatefulSet it to be scaled up to
   - **`<namespace>`** is the namespace that the StatefulSet is in: this is either the namespace (project) that your IBM Cloud Pak for AIOps installation is deployed in, or **`ibm-common-services`**.

   For example,

   ```
   oc scale statefulsets icp-mongodb --replicas=1 -n ibm-common-services
   ```

2. Check whether all pods are coming up correctly.

IBM Cloud Pak for AIOps backup and restore

Learn how to back up and restore IBM Cloud Pak for AIOps, such as for recovering from a disaster like a complete data center outage.

- Overview
- Planning
- Backup process
- Restore process
- Components

Overview

IBM Cloud Pak for AIOps backup and restore is based on the storage and database backup of critical datastores, cluster resources, and component data and settings. With the backup and restore feature, you can recover your services to a previous point in time backup to recover from data corruption, system failures, or user errors.

You can also use the backup and restore feature as a means of copying data from one cluster to another, or from one environment into another, such as for disaster recovery purposes.

If you are using backup data to restore environments on a new cluster in another data center, your recover time objective (RTO) and recovery point objective (RPO) must be sufficiently large for this usage.

IBM Cloud Pak for AIOps uses the *OpenShift APIs for Data Protection* (OADP) and the open source tool Velero to backup data to object storage. In the case of a failure, the APIs and Velero are used to restore data from the backups.

For more information about these tools, see:

- OpenShift APIs for Data Protection (OADP) features
- Velero documentation

Planning for backup and restore

When you are planning your IBM Cloud Pak for AIOps backup and restore strategy, consider what hardware requirements are needed for backing up your environment:

- You need to install the backup and restore tools in the Red Hat OpenShift Container Platform cluster. These tools include the OpenShift APIs for Data Protection (OADP), Velero, and Restic. You need to configure the OpenShift APIs and Velero with the appropriate object storage for storing your backups. This object storage must support RWX (read-write-execute) mode.
- You can backup both smaller starter and larger production sized cluster deployments.
- You can use backed up data to restore data to the existing cluster where the data was backed up, or to a new cluster.
- A restored cluster should have the same size and high availability level as the original cluster.
- Data is copied and stored in object Storage as part of the backup process. During a restoration, the same data is fetched and restored.

Notes:

- The backup and restore processes can take time. Using a backup to restore data or configurations is most useful when your recover time objective (RTO) and recovery point objectives (RPO) are met.
- As your data grows, the size of your backup storage might need to grow.
- A backup and restore of Red Hat OpenShift Container Platform or ETCD is not completed as part of the IBM Cloud Pak for AIOps backup and restore feature.
- Configure the duration between your backups to be less than your RPO. By default, the RPO is 12 hours, which requires backups to run once every 12 hours. The default RTO is 4 hours.
- The Flink state for log anomaly is not backed up and any training is lost during a failure or disaster. You need to run the training again after you recover from the disaster.
- If you upgrade your deployment, you must setup back and restore again. For more information, see Upgrading IBM Cloud Pak for AIOps backup and restore artifacts.

Backup process

The IBM Cloud Pak® for AIOps backup process is capable of backing up data for both IBM Cloud Pak for AIOps and Infrastructure Automation. The backup of data is completed by backing up the datastores that include IBM Cloud Pak for AIOps data, and if available, Infrastructure Automation data. Some datastores are backed up and restored in their entirety. Other datastores are backed up and restored depending on a particular component need.

The following table shows the IBM Cloud Pak® for AIOps datastores and persistent volume claims (PVCs) that are backed up:

Table. Backed up datastores

Datastores	Backup type	Backup PVC name	Access mode
`CouchDB`	Volume backup	data-c-example-couchdbcluster-m-0	ReadWriteOnce
`Cassandra`	Data export	back-aiops-topology-cassandra-0	ReadWriteOnce
`EDB Postgres`	Data export	postgres-backup-data	ReadWriteOnce
`Minio`	Volume backup	export-aimanager-ibm-minio-0	ReadWriteOnce
`ElasticSearch`	Partial (Elastic native volume backup used)	iaf-system-elasticsearch-es-snap-es-backup-pvc	ReadWriteMany
`Kafka`	None		
`Flink`	None		
`mongodb`	Data export	my-mongodump	ReadWriteOnce
`metastore`	Data export	metastore-backup-data	ReadWriteOnce
Other resources like `Tunnel` custom resource	Volume export	other-resources-backup-data	ReadWriteOnce
`CAM`	Volume export	cam-mongo-pv	ReadWriteOnce
`Infrastructure Management`	Data export	postgresql	ReadWriteOnce

The following table shows the backed up Kubernetes resources:

Table. Backed up resources

Backup resource name	Resource type
dummy-db	pod
es-backup	pod
backup-back-aiops-topology-cassandra	pod
backup-data-c-example-couchdbcluster-m	pod
backup-export-aimanager-ibm-minio	pod
backup-metastore	pod
backup-postgres	pod
backup-other-resources	pod
backup-cam	pod
aiops-ir-core-model-secret	secret
aimanager-ibm-minio-access-secret	secret
zen-secrets-aes-key	secret
BundleManifest	custom resource
ConnectorConfiguration	custom resource
ConnectorSchemas	custom resource

The backup process is run by custom backup-jobs. When the backup job runs, pods can be scaled down to ensure consistent backups are created. The backup job calls Velero to back up Kubernetes resources and volumes. Native backup scripts are run to back up datastores. The backup job then scales up pods to match their state before the backup process began. Datastores that have a selective backup types are backed up only when the datastore is updated since the previous backup.

For more information about the backup process, see:

- Backing up IBM Cloud Pak for AIOps data.
- Backing up Infrastructure Automation data.

Notes: For Elasticsearch, the backup and restore process completes incremental backup snapshots of Elasticsearch. While creating the snapshot, Elasticsearch analyses the list of the index files that are stored in the repository that is being backed up

and copies only the files that were created or changed since the last snapshot occurred. This process allows you to have multiple snapshots stored in a compact form.

Component data backups

When the backup process runs for IBM Cloud Pak for AIOps, data for the following components are backed up. This data includes cluster resources, such as Secrets, Pods, CRDs, CRs, PVCs, and PVs.

- **AI model management**

  log anomaly detection and metric anomaly detection templates and models are backed up. In addition, model and training metadata, secrets, and configurations are backed up. If this data is not backed up and is lost during a disaster, you cannot run training until you collect training data again. In addition your models and templates can be lost, which can prevent principal component analysis (PCA) from running until you run training to rebuild the models. You might not be able to recreate historical model if data loss occurs.

  For similar ticket analysis, the trained models are backed up.

  Notes

  - Component data, such as for log anomaly detections or similar ticket analysis, is not encrypted when it is backed up.
  - Raw data and data-in-transit, such as Kafka topics, are not backed up.
  - logtrain indices and other training data is not backed up other than training metadata.
  - Before you run the backup process, stop all training that is in progress. Do not start or restart training until the backup process is completed
- **Change risk**

  For change risk, risk models, risk assessments, incidents, and ticket data assets are backed up.

  Notes

  - Data-in-transit is not backed up.
  - If change or incident ticket data loss occurs, you might need to run data collection again from your ticketing system.
  - In change risk model loss occurs, you need to recreate the lost models by using the IBM Cloud Pak for AIOps IBM Cloud Pak Automation console. Historical models can be difficult to re-create.
  - In the case of risk assessments loss, there is no way to reconstruct the assessments.
- **Automations** (Runbooks)

  For Automations, runbooks, automations, execution records, and configurations are backed up.

- **Connections**

  Connector CRs, Secrets, and configurations, are backed up.

- **Event & Topology analytics**

  Probable cause data, Topology data, configurations and indexed data, is backed up.

- **Policy management**

  For policy management, operational policies and policy run time statistics (last run, count of executions) are backed up.

- **Secure Tunnel**

  Tunnel CRs, which represent connections, application mappings, and templates are backed up.

Restore process

To run the restore process, you first need to create a new cluster and install the required prerequisites on the cluster. The prerequisites include installing the required CLI tools, creating the storage classes, and updating the `bcdr/common/aiops-`

`config.json` and `bcdr/restore/restore-data.json` files with the required configuration values. Then you need to configure Velero to point to the same s3 storage where the backup is located.

You then need to create the required namespaces for IBM Cloud Pak® for AIOps by running the `restore-namespace.sh` job. This script creates only the namespaces and namespace metadata, not the contents of the namespace. This namespace restore is needed because the metadata contains SELinux settings that must match the settings from the corresponding namespaces on the cluster where you took the backup.

Then, you need to complete the installation of IBM Cloud Pak for AIOps and create an IBM Cloud Pak for AIOps instance. With the instance created and Velero configured, you can run the Velero restore jobs to populate the instance. When the jobs complete you need to run any other post restore tasks that you need to complete.

Note: Velero does not support the overriding of resources during the restore. The restoration of resources already existing in the cluster are skipped. If this occurs, sync the existing resources and create only the resources that were not restored instead of creating a new resources Operator.

For more information about the restore process, see Restoring IBM Cloud Pak for AIOps.

Procedures for backing up and restoring IBM Cloud Pak for AIOps

- Installing the backup and restore tools
- Backing up IBM Cloud Pak for AIOps
- Restoring IBM Cloud Pak for AIOps
- Upgrading IBM Cloud Pak for AIOps backup and restore artifacts

Installing the backup and restore tools

Learn how to install the tools that are required for the backup and restore of IBM Cloud Pak for AIOps.

You need to install the Red Hat OpenShift APIs for Data Protection (OADP). Velero and Restic are installed by using the OADP operator. OADP provides default Velero plug-ins that are integrated with storage providers to support backup operations.

Note: The backup and restore tools for IBM Cloud Pak for AIOps are the same as the tools for backing up and restoring Infrastructure Automation. If you have the tools installed for Infrastructure Automation, you do not need to complete the following procedure.

Prerequisites

Ensure that the following tasks are completed before you begin to install the backup and restore tools:

- Provision the object storage (S3 bucket) for storing your backups. The backup process moves backup files to an S3 bucket, which must be in an S3 compatible location, such as Amazon Web Services (AWS).
- Ensure that the following command line tools are installed on your workstation:
  - `kubectl`
  - `oc`
  - `jq`
  - `git`

Procedure

1. Install the Velero client CLI
2. Install OpenShift API for Data Protection (OADP) and set environment variables
3. Download the latest ubi9 image
4. Build and push backup and restore image

1. Install the Velero client CLI

You need to install the Velero client CLI on the workstation that you plan to use to run the scripts for backing up and restoring the cluster.

1. Run the following command to download the Velero CLI client:

   ```
   wget https://github.com/vmware-tanzu/velero/releases/download/v1.10.3/velero-v1.10.3-linux-amd64.tar.gz
   ```

2. Extract the tarball:

   ```
   tar -xvf velero-v1.10.3-linux-amd64.tar.gz -C /tmp
   ```

3. Move the extracted velero binary to **/usr/local/bin**:

   ```
   sudo mv /tmp/velero-v1.10.3-linux-amd64/velero /usr/local/bin
   ```

4. Verify the installation:

   ```
   velero version
   Client:
      Version: v1.10.3
      Git commit: 80cc81b540a738869546890a43255c5f9ea454ae
   ```

2. Install Red Hat OpenShift APIs for Data Protection (OADP) and set environment variables

Follow the Red Hat OpenShift documentation to install OpenShift API for Data Protection: About installing OADP).

As a convenience, if you are using S3 buckets on AWS, you can use the installation script **install-velero.sh** in this section to simplify the process.

1. Clone the IBM Cloud Pak for AIOps samples GitHub repository that includes the backup and restore scripts. Clone this repository by running the following command:

   ```
   git clone https://github.com/IBM/cp4waiops-samples.git
   ```

2. Update the install-velero-config.json file.

   1. Go to the directory **<Path>/4.2.0/install** where you have the backup and recovery files downloaded. **<Path>** is the path to where you downloaded and extracted the IBM Cloud Pak for AIOps backup and restore files.

   2. Open the **install-velero-config.json** file for editing. Configure the following settings in the file:

      - **aws_access_key_id** - Access key ID to connect to your object storage S3 bucket.
      - **aws_secret_access_key** - Secret access key to connect to your S3 bucket.
      - **bucket_name** - Name of the S3 bucket where backup data will be stored.
      - **bucket_region** - Region where the S3 bucket is deployed.
      - **namespace** - Namespace name in which you want to deploy Velero. If the namespace does not exist, it is created when you run the Velero installation script.
      - **backup_label** - Label for creating a folder structure to organize sets of backups together.

3. Log in to the Red Hat OpenShift Container Platform cluster by running the following command:

   ```
   oc login --token=<TOKEN> --server=<URL>
   ```

   Where:

   - **<TOKEN>** is the token that you use to log in to the Red Hat OpenShift Container Platform cluster.
   - **<URL>** is the Red Hat OpenShift Container Platform server URL.

4. Change to the **install** directory where the install script is located:

   ```
   cd <Path>/bcdr/4.2.0/install
   ```

Where **<Path>** is the path to where you downloaded and extracted the IBM Cloud Pak for AIOps backup and restore files.

5. Run the OADP installation script:

   ```
   nohup ./install-velero.sh > install-velero.log &
   ```

6. Check the logs by running the following command:

   ```
   tail -f install-velero.log
   ```

7. Check the Velero pods status by running the following command:

   ```
   oc get pods -n <veleroNamespace>
   ```

   Where **<veleroNamespace>** is the namespace where Velero is installed.

   The Velero and Restic pods should be in a running state.

8. Check the status of the **backupStorageLocation** by running the following command:

   ```
   oc get backupStorageLocation -n <veleroNamespace>
   ```

   Where **<veleroNamespace>** is the namespace where Velero is installed.

   The **backupStorageLocation** needs to be in an available status.

9. Set environment variables.

   ```
   export IMAGE_REGISTRY_URL=<Image Registry Server URL>
   export REPOSITORY=<Repository>
   export USERNAME=<USERNAME>
   ```

   Where:

   - **<Image Registry Server URL>** is the image registry server URL.
   - **<Repository>** is the repository where you put the image.
   - **<USERNAME>** is the username for logging in to the image registry server.

   Note: You can use any image registry server to store the backup image, such as quay.io, Docker Registry, or any other private image registry. You must have image pull and push access to the image registry server. The image repository must also be accessbile to the cluster where you plan to run the backup and restore processes.

3. Download the latest ubi9 image

1. Download the latest **ubi9** image from Docker.

2. Run the following command to login to your registry.

   ```
   docker login ${IMAGE_REGISTRY_URL} -u ${USERNAME}
   ```

3. Tag the **ubi19** image.

   ```
   docker tag redhat/ubi9:latest ${IMAGE_REGISTRY_URL}/${REPOSITORY}/ubi9:latest
   ```

4. Push the **ubi9** image to your repository.

   ```
   docker push ${IMAGE_REGISTRY_URL}/${REPOSITORY}/ubi9:latest
   ```

5. Update the YAML files that you cloned to reference the new **ubi9** image.

   Update **spec.image** to be **<registry>/ubi9:latest** in the following YAML files, where **<registry>** is **${IMAGE_REGISTRY_URL}/${REPOSITORY}**

   - backup/scripts/cam/cam-backup.yaml
   - backup/scripts/common-services/dummy-db.yaml
   - backup/scripts/elasticsearch/es-backup-pod.yaml
   - backup/scripts/metastore/metastore-backup.yaml
   - backup/scripts/other-resources/other-resources-backup.yaml

- backup/scripts/postgres/postgres-backup.yaml
- backup/scripts/sts-backup-pod-template.yaml
- restore/common-services/cleanup-pod.yaml

4. Build and push backup and restore image

1. Build the Docker image.
    1. Change to the directory where you have the backup and recovery files downloaded.

       ```
       cd <Path>/bcdr/4.2.0/
       ```

       Where **<Path>** is the path to where you downloaded and extracted the IBM Cloud Pak for AIOps backup and restore files.

    2. Build the **cp4waiops-bcdr** Docker image by running following command:

       ```
       docker build -t cp4waiops-bcdr:v4.2.0 -f Dockerfile .
       ```

2. Tag and push the Docker image to the image registry.
    1. Run the following command:

       ```
       docker tag cp4waiops-bcdr:v4.2.0 ${IMAGE_REGISTRY_URL}/${REPOSITORY}/cp4waiops-bcdr:v4.2.0
       ```

    2. Run the following command:

       ```
       docker push ${IMAGE_REGISTRY_URL}/${REPOSITORY}/cp4waiops-bcdr:v4.2.0
       ```

Troubleshooting

- Red Hat OpenShift APIs for Data Protection (OADP) installation failed
- Velero backups are not showing after installation

Red Hat OpenShift APIs for Data Protection (OADP) installation failed

If your installation of OADP fails for any reason, delete any related files and reattempt the installation.

For example, the install script **install-velero.sh** might not complete successfully. If this occurs an error similar to the following message can display:

```
error: unable to recognize "STDIN": no matches for kind "DataProtectionApplication" in version "oadp.openshift.io/v1alpha1"
```

This can occur if the network or cluster is running slow and taking too long to pull the required images. This delay causes the OADP operator to take too long to create the required custom resource.

If the preceding issue occurs, or the install fails for another reason, run the following **oc** commands to delete any existing Velero installation files before attempting to install again. These commands use **<veleroNamespace>** as the project (namespace) where Velero is installed. Replace this value with the project where you are attempting to install Velero.

1. Delete any cluster service versions (CSV) in the project:

   ```
   oc delete csv --all -n <veleroNamespace>
   ```

2. Delete any subscriptions in the project:

   ```
   oc delete subscription --all -n <veleroNamespace>
   ```

3. Delete any deployments in the project:

   ```
   oc delete deploy --all -n <veleroNamespace>
   ```

4. Delete any daemonsets in the project:

   ```
   oc delete ds --all -n <veleroNamespace>
   ```

5. Delete the project (namespace):

   ```
   oc delete project <veleroNamespace>
   ```

Any existing installation is deleted and you can reattempt the installation.

Velero backups are not showing after installation

If your Velero backups do not show, complete the following steps:

1. Correct the **bucket** and **prefix** property within the **backupstoragelocation** resource that is named **bcdr-s3-location**. The **bucket** and **prefix** name should match the value that is defined within the **install-velero-config.json** file during the Velero installation.

   Run the **oc edit** command to edit the **backupstoragelocation** resource:

   ```
   oc edit backupstoragelocation bcdr-s3-location -n <veleroNamespace>
   ```

   Where **<veleroNamespace>** is the namespace where Velero is installed.

2. Delete the Velero pod by running the following command:

   ```
   oc delete pod <pod> -n <veleroNamespace>
   ```

   Where **<veleroNamespace>** is the namespace where Velero is installed, and **<pod>** is the name of the Velero pod.

   The backups should now be available.

3. Run the following command to list the backups:

   ```
   velero get backup
   ```

Backing up IBM Cloud Pak for AIOps data

Learn how to back up data for IBM Cloud Pak for AIOps components.

The backup process uses Velero for cluster object backups and uses Restic for volume backups. The backup script must be the same version as your deployment.

- Prerequisites
- Procedure
- Troubleshooting

Note: If you also need to back up data for Infrastructure Automation components, the process and tools to use are the same. The only change that is required is when you are packaging the Helm chart. During this step you need to configure the value of the **enabledComponents** parameter to include **'"IA"'** as well as **'"AIOPS"'**. All other steps remain the same.

Prerequisites

- Provision S3 compliant object storage for your backups. You must have sufficient storage to backup the data that is contained in the datastores and persistent volume claims (PVCs) of the components that are backed up, as detailed in Backup process. The storage must also be accessible by any environment that a backup might need to be restored to. The backup process moves backup files to an S3 bucket, which must be in an S3 compatible location. For instance, you can use the following public or private cloud options for provisioning storage for your backups:
  - MinIO
  - Red Hat OpenShift Data Foundation object store
  - IBM Cloud Object Storage
  - AWS Cloud

  Note: As your data grows, the size of your backup storage might need to grow, and you must ensure that sufficient storage for backups is provisioned to accommodate any growth in data size.

- If they are not already installed, install the backup and restore tools. You need to install the Red Hat OpenShift APIs for Data Protection (OADP), Velero, and Restic in the Red Hat OpenShift Container Platform cluster. For more information, see Installing the backup and restore tools.

  Important: Ensure that the OpenShift APIs for Data Protection (OADP) and Velero are configured to point to the same object storage (S3 bucket) that includes the backup that you plan to use.

- Before you run the backup process, stop all training that is in progress. Do not start or restart training until the backup process is completed

Procedure

1. Clone the IBM Cloud Pak for AIOps samples GitHub repository to retrieve the latest backup and restore scripts. Clone this repository by running the following command:

   ```
   git clone https://github.com/IBM/cp4waiops-samples.git
   ```

2. Log in to the Red Hat OpenShift Container Platform cluster by running the following command:

   ```
   oc login --token=<TOKEN> --server=<URL>
   ```

   Where:

   - **<TOKEN>** is the token that you use to log in to the Red Hat OpenShift Container Platform cluster.
   - **<URL>** is the Red Hat OpenShift Container Platform server URL.

3. Create an image pull secret by running the following command:

   ```
   oc create secret docker-registry backup-secret -n <veleroNamespace> --docker-server=
   <Image Registry Server URL> --docker-username=<USERNAME> --docker-password=<PASSWORD> --
   docker-email=<EMAIL>
   ```

   Where:

   - **<Image Registry Server URL>** is the image registry server URL.
   - **<veleroNamespace>** is the velero namespace.
   - **<USERNAME>** is the username to log in to the image registry server.
   - **<PASSWORD>** is the password to log in to the image registry server.
   - **<EMAIL>** is the email for image registry server.

4. Package the Helm Chart

   1. Change to the directory where you have the backup and recovery files downloaded.

      ```
      cd <Path>/bcdr/4.2.0/backup
      ```

      Where **<Path>** is the path to where you downloaded and extracted the IBM Cloud Pak® for AIOps backup and restore files.

   2. Update the following parameters in the **values.yaml** file. The file is located in the **./helm** directory.

      - **repository** - The name of the Docker image, as created in Build and push Docker image for backup and restore.

        For example, **xy.abc.io/cp4waiops/cp4waiops-bcdr**. In this example, **xy.abc.io** is the image registry server URL, **cp4waiops** is the name of the repository and **cp4waiops-bcdr** is the name of the Docker image.

      - **pullPolicy** - The policy for determining when to pull the Docker image from the image registry server.

        For example, to force pull the image, use the **Always** policy.

      - **tag** - The tag for the Docker image. That is, **4.2.0** (as created in Build and push Docker image for backup and restore).

      - **pullSecret** - The name of the image pull secret.

- **schedule** - (optional when on-demand backups are activated) The Cron expression for the automated backup. For example, to take a backup once a day, use the `0 0 * * *` Cron expression.
- **backupStorageLocation** - The **velero** storage location where backed up data is stored. For example, **bcdr-s3-location**.

  Use the **oc get backupstoragelocation -n <veleroNamespace>** command to get the **backupStorageLocation** on the OpenShift cluster.
- **backupNameSuffix** - The prefix for the backup name when the backup is created by using the job. Generally, it can be name of source cluster itself. For example, **aiops-cluster-backup-106**.
- **aiopsNamespace** - The namespace where IBM Cloud Pak for AIOps is installed.
- **csNamespace** - The namespace where IBM Cloud Pak foundational services is installed.
- **veleroNamespace** - The namespace where Velero is installed.
- **ttl** - The time-to-live setting for the backup. The backed up data is retained until the **ttl** value is reached (expires). For example **720h0m0s**.
- **enabledNamespaces** The namespaces where components that need to be backed up are installed.

  For example, the **ibm-common-services** namespace represents the IBM Cloud Pak foundational services component. You can delete any unused namespaces from the list to reduce the time that is required for the backup process. You can update the list as shown if you have installed only two components, for example, IBM Cloud Pak foundational services and IBM Cloud Pak for AIOps

  ```
  enabledNamespaces:
  - '"ibm-common-services"'
  - '"cp4waiops"'
  ```

  Component: IBM Cloud Pak foundational services

  - Namespaces: **ibm-common-services**

  Component: IBM Cloud Pak for AIOps

  - Namespaces: **cp4waiops**
- **enabledComponents** - Required. The list of components to back up.

  The backup and restore processes support backing up and restoring both the IBM Cloud Pak for AIOps and Infrastructure Automation components. These components can be installed together or independtly. You can specify to back up both, or only one of these components. Any other value is ignored and a corresponding error message is generated.

  To specify Infrastructure Automation, set the value `'"IA"'`.

  To specify IBM Cloud Pak for AIOps, set the value `'"AIOPS"'`.

  The following configuration shows both components listed:

  ```
  enabledComponents:
  - '"IA"'
  - '"AIOPS"'
  ```

3. Package the Helm Chart.

   ```
   helm package ./helm
   ```

5. Trigger an automated backup

   1. Change to the **backup** directory where the script is located:

      ```
      cd <Path>/bcdr/4.2.0/backup
      ```

      Where **<Path>** is the path to where you downloaded and extracted the IBM Cloud Pak for AIOps backup and restore files.

2. Deploy the backup job by running the following command:

   ```
   helm install backup-job clusterbackup-0.1.0.tgz
   ```

6. On-demand backups

   The deployment of an automated backup job is a prerequisite for the on-demand job. Only after you initiate an automated backup job, then you can trigger an on-demand backup.

   1. Deploy the on-demand backup job by running the following command:

      ```
      oc create job --from=cronjob/backup-job on-demand-backup-job -n <veleroNamespace>
      ```

      Where:

      - **veleroNamespace** - The namespace where Velero is installed.
   2. Check the on-demand backup pods status by running the following command:

      ```
      oc get pods -n <veleroNamespace>
      ```

   3. Check the on-demand backup job logs by running the following command:

      ```
      oc logs -f <on-demand-backup-job-***> -n <veleroNamespace>
      ```

   4. Check the backup status by running the following command:

      ```
      velero get backup <BACKUP_NAME> -n <veleroNamespace>
      ```

      Where:

      - **veleroNamespace** is the namespace where Velero is installed.
      - **<BACKUP_NAME>** is the name of the Backup. You can see the backup name after the on-demand backup job is complete. For example, you might see the backup name **aiops-cluster-backup-106-1622193915** in the on-demand backup job log as follows:

        ```
        Waiting for backup aiops-cluster-backup-106-1622193915 to complete
        ```

        The status is in **completed** state when the backup is finished, such as in the following sample output:

        ```
        aiops-backup-1654171690-testfest34   Completed                   2022-06-02
        05:08:11 -0700 PDT   29d       bcdr-data-protection-app-1
        cp4aiops.ibm.com/backup=t
        ```

   5. Review the backup result summary that is included within the logs of the backup job.

      Note: A **backup-result-config** ConfigMap is created to include this result. To view this ConfigMap, run the following command:

      ```
      oc get cm backup-result-config -n <namespace> -o yaml
      ```

      Where **<namespace>** is the project (namespace) where IBM Cloud Pak for AIOps is installed.

   6. Optional - update the **schedule** parameter in the **values.yaml** file (located in the **./helm** directory). The Cron expression for the automated backup. For example, to take a backup once a day, use the **0 0 * * *** Cron expression.

7. Monitor the backup Job

   1. Check the backup pods status by running the following command:

      ```
      oc get pods -n <veleroNamespace>
      ```

      Where:

      - **veleroNamespace** - The namespace where Velero is installed.
   2. Check the backup job logs by running the following command:

      ```
      oc logs -f <backup-job-***>
      ```

3. Check the backup status by running the following command:

   ```
   velero get backup <BACKUP_NAME> -n <veleroNamespace>
   ```

   Where:

   - **veleroNamespace** is the namespace where Velero is installed.
   - **<BACKUP_NAME>** is the name of the Backup. You can see the backup name after the backup job is complete. For example, you might see the backup name **aiops-cluster-backup-106-1622193915** in the backup job log as follows:

     ```
     Waiting for backup aiops-cluster-backup-106-1622193915 to complete
     ```

Troubleshooting

- Velero backups are stuck in the *In progress* state
- Classifier and layout pods remain not ready after running a backup
- Helm install backup job command failed

Velero backups are stuck in the *In progress* state

If your Velero backups remain stuck in the *In progress* state for an unexpected duration, complete the following steps. This procedure stops the backup process so that you can try the backup again.

1. Delete the Velero pod by running the following command:

   ```
   oc delete pod <pod> -n <veleroNamespace>
   ```

   Where **<veleroNamespace>** is the namespace where Velero is installed, and **<pod>** is the name of the Velero pod.

2. Delete the backup that is stuck in the *In progress* state:

   ```
   velero delete backup <backup> -n <veleroNamespace>
   ```

   Where:

   - **<backup>** is the backup that you want to delete. The backup process should begin again.
   - **veleroNamespace** is the namespace where Velero is installed.
3. Wait for the process to complete and verify that the backup is created.

Classifier and layout pods remain not ready after running a backup

After you complete a backup, you might notice that the classifier (**aiops-ir-analytics-classifier**) and layout (**aiops-topology-layout**) pods do not change to a ready state, which can cause **AIOpsAnalyticsOrchestrator** to be stuck in an updating phase. This issue can occur after most backups. To resolve this issue, you need to either restart the pods or increase the resource limits for the pods.

- For the classifier (**aiops-ir-analytics-classifier**) pod, restart the pod.
- For the layout (**aiops-topology-layout**) pod, restart the pod. If the restart does not result in the pod becoming ready, increase the resource limits for the pod. Doubling the **cpu** and **memory** limits and request allocations for the pods can result in the pod starting and entering a ready state.

Helm install backup job command failed

When you are running the **helm install backup-job clusterbackup-0.1.0.tgz** command, you might encounter the command failing with an error that is similar to the following error:

```
Error: admission webhook "trust.hooks.securityenforcement.admission.cloud.ibm.com" denied the
request:
Deny "<Image Registry Server URL>/<Repository>/cp4waiops-bcdr:latest", no matching
repositories in ClusterImagePolicy and no ImagePolicies in the "veleroNamespace" namespace
```

Where:

- **<Image Registry Server URL>** is the image registry server URL.

- `<Repository>` is the repository where you put the image.

If you encounter this error, complete the following steps to resolve the issue:

1. Uninstall the `backup-job` job by running the following command:

   ```
   helm uninstall backup-job -n <veleroNamespace>
   ```

   Where:

   - `veleroNamespace` is the namespace where Velero is installed.
2. Create a `backup-image-policy.yaml` file and add the following content within the file:

   ```
   apiVersion: securityenforcement.admission.cloud.ibm.com/v1beta1
   kind: ClusterImagePolicy
   metadata:
     name: backup-image-policy
   spec:
    repositories:
     - name: "<Image Registry Server URL>/<Repository>/cp4mcm-bcdr:latest"
       policy:
   ```

3. Apply the policy by running the following command:

   ```
   oc apply -f backup-image-policy.yaml
   ```

4. Deploy the backup job by running the following command:

   ```
   helm install backup-job clusterbackup-0.1.0.tgz
   ```

Restoring IBM Cloud Pak for AIOps data

Learn how to restore data for IBM Cloud Pak for AIOps components to a cluster, such as for disaster recovery.

The following procedure restores all backed up data that exists in the specified backup for IBM Cloud Pak for AIOps components. The steps in the following procedure restore IBM Cloud Pak for AIOps in a new cluster.

Notes:

- All required storage classes must be created prior to running the restore process. The storage classes must have the same name as the backup cluster.
- You can restore a backup only within an environment that has the same version of IBM Cloud Pak for AIOps as the environment where the backup was created. For example, a backup of an IBM Cloud Pak for AIOps 4.2.0 environment must be restored within a cluster that has IBM Cloud Pak for AIOps 4.2.0 installed. If you need to upgrade as well as restore data, complete the restore process before you upgrade.
- If you are also restoring Infrastructure Automation data, the overall procedure is the same with the following additional steps required:
  - After you install IBM Cloud Pak for AIOps on the cluster where you are restoring data, you need to install Infrastructure Automation.
  - After you restore your IBM Cloud Pak for AIOps data, you need to restore your Infrastructure Automation data.

  For more information, see Restoring Infrastructure Automation.

Procedure

The process for restoring IBM Cloud Pak for AIOps to a cluster involves the following step sections:

1. Set up your new cluster for backup and restore.

   Your first steps for restoring a cluster require you to install OpenShift, configure storage, and install the backup and restore tools.

2. Prepare the backup data for restoring.

   Before you can run any restore jobs, you need to verify your backed up data and prepare the data for restoring.

3. Restore the cluster namespaces and install IBM Cloud Pak for AIOps

   Since the restore job does not install IBM Cloud Pak for AIOps, you need to first install IBM Cloud Pak for AIOps before you can run the restore jobs for restoring database and component data. A restore job for restoring the cluster namespaces is available and must be run before you install IBM Cloud Pak for AIOps.

4. Restore the IBM Cloud Pak for AIOps data

   This procedure restore the data for all backed up IBM Cloud Pak for AIOps databases and components. Optionally, you can choose to restore only individual components. For details, see Restoring individual components.

5. Optional. Restore Infrastructure Automation data

6. Post-restore tasks

If you encounter any issues with the restore process, see Troubleshooting

1. Set up your new cluster for backup and restore

1. Install OpenShift and create a cluster using the instructions in Installing Red Hat OpenShift Container Platform.

   IBM Cloud Pak for AIOps requires OpenShift to be installed and running. You must have administrative access to your OpenShift cluster.

   Important: Ensure that the version of Red Hat OpenShift Container Platform that you install is the same as the version that was installed in the backed up cluster.

   For information on supported versions of OpenShift, see Supported Red Hat OpenShift Container Platform versions. The hardware architecture that you install IBM Cloud Pak for AIOps on must be AMD64.

   Note: IBM Cloud Pak for AIOps uses the OpenShift image registry when it builds images in real time. If the OpenShift image registry is not persistent and the registry respawns, then workloads can temporarily fail until respawning is complete. A persistent OpenShift image registry is recommended to avoid this. For more information, see Setting up and configuring the registry in the Red Hat OpenShift Container Platform documentation.

2. Install the OpenShift command line interface (`oc`) on your cluster's boot node and run `oc login`, using the instructions in Getting started with the OpenShift CLI.

3. Configure storage

   The storage configuration must satisfy your sizing requirements. For more information on the storage classes that are needed for installing IBM Cloud Pak for AIOps, see Storage.

   Important: All required storage classes must be created prior to running the restore process. The storage classes must have the same name as the backup cluster.

4. Install the backup and restore tools

   1. Install the Red Hat OpenShift APIs for Data Protection (OADP), Velero, and Restic in the Red Hat OpenShift Container Platform cluster. For more information, see Installing the backup and restore tools.

      Important: Ensure that the OpenShift APIs for Data Protection (OADP) and Velero are configured to point to the same object storage (S3 bucket) that includes the backup that you plan to use.

2. Prepare the backup data for restoring

1. Check the backup status

   Check the backup status to ensure that the backup that you want to restore in your cluster is complete. Run the following command to check the contents of the backup:

   ```
   velero describe backup <backup-name> --details
   ```

The output should list the backed-up data for IBM Cloud Pak for AIOps (**cp4waiops/***) and for IBM Cloud Pak foundational services (**ibm-common-services/***). If you also backed up Infrastructure Automation data, this data (**cp4waiops/***) should also be listed.

The output should resemble the following sample output:

```
v1/PersistentVolumeClaim:
  - cp4waiops/back-aiops-topology-cassandra-0
  - cp4waiops/data-c-example-couchdbcluster-m-0
  - cp4waiops/export-aimanager-ibm-minio-0
  - cp4waiops/iaf-system-elasticsearch-es-snap-es-backup-pvc
  - cp4waiops/postgres-backup-data
  - cp4waiops/metastore-backup-data
  - ibm-common-services/my-mongodump
v1/Pod:
  - cp4waiops/backup-back-aiops-topology-cassandra-0
  - cp4waiops/backup-data-c-example-couchdbcluster-m-0
  - cp4waiops/backup-export-aimanager-ibm-minio-0
  - cp4waiops/backup-metastore
  - cp4waiops/es-backup
  - cp4waiops/backup-metastore
  - ibm-common-services/dummy-db
v1/Secret:
  - cp4waiops/aimanager-ibm-minio-access-secret
  - cp4waiops/aiops-ir-core-model-secret
  - ibm-common-services/icp-mongodb-admin
  - ibm-common-services/icp-serviceid-apikey-secret

Velero-Native Snapshots: <none included>

Restic Backups:
 Completed:
  cp4waiops/backup-back-aiops-topology-cassandra-0: backup
  cp4waiops/backup-data-c-example-couchdbcluster-m-0: backup
  cp4waiops/backup-export-aimanager-ibm-minio-0: backup
  cp4waiops/backup-metastore: data
  cp4waiops/es-backup: elasticsearch-backups
  cp4waiops/backup-postgres: backup
  ibm-common-services/dummy-db: mongodump
```

2. Create an image pull secret by running the following command. This is to pull the Docker image for backup and restore, which is created in Build and push Docker image for backup and restore:

   ```
   oc create secret docker-registry restore-secret -n velero --docker-server=<Image Registry Server URL> --docker-username=<USERNAME> --docker-password=<PASSWORD> --docker-email=<EMAIL>
   ```

   Where:

   - **<Image Registry Server URL>** is the image registry server URL.
   - **<USERNAME>** is the username to log in to the image registry server.
   - **<PASSWORD>** is the password to log in to the image registry server.
   - **<EMAIL>** is the email for image registry server.
3. Package and install the Helm Chart

   1. Change to the **restore** directory where you need to package the Helm Chart:

      ```
      cd <Path>/bcdr/4.2.0/restore
      ```

      Where **<Path>** is the path to where you downloaded and extracted the IBM Cloud Pak® for AIOps backup and restore files.

   2. Update the following parameters in the **values.yaml** file. The file is located in the **./helm** directory:

      - **backupName** - The name of the backup that you are restoring.
      - **aiopsNamespace** - The namespace where IBM Cloud Pak for AIOps is installed.
      - **csNamespace** - The namespace where IBM Cloud Pak foundational services is installed.
      - **veleroNamespace** - The namespace where Velero is installed.

   3. Package the Helm Chart.

```
helm package ./helm
```

4. Install the Helm Chart for restoring data by running the following job:

```
helm install restore-job clusterrestore-0.1.0.tgz
```

3. Restore the cluster namespaces and install IBM Cloud Pak for AIOps

1. Update the image and job names

    1. Change to the **restore** directory where the restore script is located:

    ```
    cd <Path>/bcdr/4.2.0/restore
    ```

    Where **<Path>** is the path to where you downloaded and extracted the IBM Cloud Pak® for AIOps backup and restore files.

    2. Open the following YAML files for editing:

        - **ns-restore-job.yaml** - This file is used for the job to restore the cluster namespaces
        - **aiops-restore-job.yaml** - This file is used for the job to restore IBM Cloud Pak for AIOps
        - Optional. **ia-restore-job.yaml** - This file is used for the job to restore Infrastructure Automation. If you do not need to restore Infrastructure Automation data, you can skip updating this file.

    3. Update the image name in each YAML file. You must specify the name of the image that you have pushed to the image repository: **<Image Registry Server URL>/<Repository>/cp4waiops-bcdr:4.2.0**.

    4. Update **JOB_NAME** value in each YAML file:

        - For the **ns-restore-job.yaml** file, update the value to be **ns-restore-job**.
        - For the **aiops-restore-job.yaml** file, update the value to be **aiops-restore-job**.
        - Optional. For the **ia-restore-job.yaml** file, update the value to be **ia-restore-job**.

    5. Save your changes.

2. Restore the cluster namespaces

    You need to restore the projects (namespaces) of the backed up cluster so that your new cluster includes the metadata with the SELinux settings that need to match the settings for the backup data that you plan to restore.

    These steps only restore the namespaces and namespace metadata. The commands do not restore the contents of the namespaces.

    1. Change to the **restore** directory where the restore script is located:

    ```
    cd <Path>/bcdr/4.2.0/restore
    ```

    Where **<Path>** is the path to where you downloaded and extracted the IBM Cloud Pak® for AIOps backup and restore files.

    2. Optional. Delete any existing namespace restore jobs:

    ```
    oc delete -f ns-restore-job.yaml
    ```

    3. Create a job to restore the cluster namespaces:

    ```
    oc create -f ns-restore-job.yaml
    ```

    4. Check the restore job logs by running the following command:

    ```
    oc logs -f <ns-restore-job-***>
    ```

    5. Check the velero-restore status for the namespace by running the following command:

    ```
    velero get restore <RESTORE_NAME>
    ```

    Where **<RESTORE_NAME>** is the name of the restore for namespace. You can see the restore name after the restore job is completed. For example, you might see the restore name **cs-namespace-restore-20221006054710** and **aiops-namespace-restore-20221006054710** within the restore job log as follows:

```
Restore request "cs-namespace-restore-20221006054710" submitted successfully.
```

and

```
Restore request "aiops-namespace-restore-20221006054710" submitted successfully.
```

Ensure that the projects (namespaces) are restored before you proceed.

3. Install IBM Cloud Pak for AIOps

   For more information, see Installing IBM Cloud Pak for AIOps.

   Important:

   - Ensure that the version of IBM Cloud Pak for AIOps that you are installing is the same as the version that was installed in the backed up cluster.
   - The backup includes keys and certificates from the backed up cluster. Ensure that your new cluster is configured to support the use of these keys and certificates so that the restored data can be accessed.
   - Wait until the installation is complete and all pods in the IBM Cloud Pak for AIOps project (namespace) are running before you proceed with the following restore steps.
4. Optional. If you also need to restore Infrastructure Automation data, you need to install the Infrastructure Automation operators and create the required custom resources.

   - For more information about installing Infrastructure Automation, see Installing Infrastructure Automation.
   - For more information about restoring Infrastructure Automation, see Restoring Infrastructure Automation.

4. Restore the IBM Cloud Pak for AIOps data

This procedure restore the data for all backed up IBM Cloud Pak for AIOps databases and components. Optionally, you can choose to restore only individual components. For details, see Restoring individual components.

1. Change to the **restore** directory where the restore script is located:

   ```
   cd <Path>/bcdr/4.2.0/restore
   ```

   Where **<Path>** is the path to where you downloaded and extracted the IBM Cloud Pak® for AIOps backup and restore files.

2. Optional. Delete any existing IBM Cloud Pak for AIOps restore jobs:

   ```
   oc delete -f aiops-restore-job.yaml
   ```

3. Create a job to restore IBM Cloud Pak for AIOps:

   ```
   oc create -f aiops-restore-job.yaml
   ```

4. Check the restore job logs by running the following command:

   ```
   oc logs -f <aiops-restore-job-***>
   ```

5. Check the velero-restore status by running the following command:

   ```
   velero get restore <RESTORE_NAME>
   ```

   Where **<RESTORE_NAME>** is the name of the restore for IBM Cloud Pak for AIOps. You can see the this restore name after the restore job is completed. or example, you might see the restore name **cassandra-restore-20221006054710** for the Cassandra restore in the restore job log as follows:

   ```
   Restore request "cassandra-restore-20221006054710" submitted successfully.
   ```

   Similarly, the velero restore name for other IBM Cloud Pak for AIOps components can display in the restore job log.

5. Optional. Restore the Infrastructure Automation data

If you are also restoring Infrastructure Automation data, run the commands to restore the Infrastructure Automation data. For more information, see Restoring Infrastructure Automation.

6. Post-restore tasks

1. Update your connections.

   If you have a connection that you are restoring, the status for these connections can be in error after the restore process completes. To resolve this status, you need to edit and save your connections with the *Data and tool connections* feature in the IBM Cloud Pak for AIOps UI IBM Cloud Pak Automation console. Editing the connection regenerates the associated Flink job for the connection, which updates the status. For more information about editing these connections, see Configuring data and tool connections.

2. Install the secure tunnel connections for any restored secure tunnel data.

   1. Optional. If the tunnel exists on your cluster, uninstall the existing tunnel connection.
   2. Install the Secure Tunnel connector on your restored cluster.
   3. Create the secure tunnel connection on your restored cluster.

3. If you have any connections with remotely deployed connectors, then you must download and run new bootstrap commands from the cluster on which the restore was run.

4. Enable authentication with the restored JWT certificate.

   1. Retrieve the new **initial_admin_password** from the **admin-user-details** secret.

      ```
      export PROJECT=<project>
      PASS=$(oc get secret admin-user-details -n ${PROJECT} -o
      jsonpath='{.data.initial_admin_password}' | base64 -d)
      ```

      Where **<project>** is the namespace (project) that your IBM Cloud Pak for AIOps subscription is deployed in.

   2. Disable the **admin** user.

      ```
      oc rsh -n ${PROJECT} $(oc get pod -l component=usermgmt -n ${PROJECT} | tail -1 |
      cut -f1 -d\ ) /usr/src/server-src/scripts/manage-user.sh --disable-user admin
      ```

   3. Enable the **admin** user with the new password.

      ```
      echo "$PASS" | oc rsh -n ${PROJECT} $(oc get pod -l component=usermgmt -n ${PROJECT}
      | tail -1 | cut -f1 -d\ ) /usr/src/server-src/scripts/manage-user.sh --enable-user
      admin
      ```

   4. Delete the secret **app-api-user-jwt**. It will be recreated automatically after a few seconds.

      ```
      oc delete secret app-api-user-jwt -n ${PROJECT}
      ```

Restoring individual components

If needed, you can choose to restore data for specific individual databases and components instead of for all databases and components at once.

1. Change to the **restore** directory where the restore script is located:

   ```
   cd <Path>/bcdr/4.2.0/restore
   ```

   Where **<Path>** is the path to where you downloaded and extracted the IBM Cloud Pak® for AIOps backup and restore files.

2. Copy the **aiops-restore-job.yaml** file. Rename your new file **<component>-restore-job.yaml**, where **<component>** is the name of the component that you are restoring.

   For example, if you are restoring Cassandra, rename the file to be **cassandra-restore-job.yaml**.

3. Open the new **<component>-restore-job.yaml** file for editing. Update the **name** and **command** sections to match the values for the component that you are restoring:

   - Update the **name** of the restore job in the metadata section to be the individual component job, such as **cassandra-restore-job**.

- Update the command section `command: ["/bin/bash", "restore.sh","-aiops"]` to replace `-aiops` with the respective argument for the component that you are restoring. For the list of component arguments, see the table that follows this procedure.

4. Create a job to restore the individual component.

   ```
   oc create -f <component>-restore-job.yaml
   ```

5. Check the restore job logs by running the following command:

   ```
   oc logs -f <cp4waiops-component-restore-job-***>
   ```

6. Check the velero-restore status by running the following command:

   ```
   velero get restore <RESTORE_NAME>
   ```

   Where **<RESTORE_NAME>** is one of the restore names for the component. You can view the names for the component in the restore job log when the restore job is completed. For example, the restore name for a Cassandra restore can be **cassandra-restore-20221006054710**, which can display in an entry similar to the following example log entry:

   ```
   Restore request "cassandra-restore-20221006054710" submitted successfully.
   ```

Component or database arguments for restore job command configuration

Table. Component or database arguments for restore job commands

Component or Database	Argument
Cassandra	**-cassandra**
CouchDB	**-couchdb**
Elasticsearch	**-es**
Metastore	**-metastore**
Minio	**-minio**
Postgres	**-postgres**
IBM Cloud Pak foundational services	**-cs**
Connection CR	**-connectioncr**
Secure Tunnel CR	**-tunnelcr**

Troubleshooting

- Stale alert data exists after restoration
- Restore process for ElasticSearch failed
- Restore fails with CouchDB pod in CrashLoopBackOff
- LDAP user login is not working after a restore
- Data is not being processed after restoration
- Applications still show non-existent active incidents and alerts notations
- Topology observer jobs do not run after restore
- Restore process terminated mid-process with partial data available
- Velero restore is stuck in an In progress state
- Helm install restore job command failed
- Cassandra restore fails with **org.apache.cassandra.io.FSReadError** exception
- Connections missing after a restore
- Netcool connection is missing details after restore
- Troubleshooting the Infrastructure Automation restore

Stale alert data exists for topology resources after restoration

After you complete a restore, you might notice notice that stale alert data exists for topology resources. For instance, if you search for a resource that had alert data, you might see outdated alert data. To clear the historical alert data, you can run the **statusClear** crawler.

You can run the crawler to clear stale events by using the topology service Swagger pages:

1. Open the Swagger API for the topology service. For more information, see Accessing Topology service Swagger UI.

2. Go to the Crawlers section.
3. Open the statusClear crawler.
4. Run the crawler with the default message body.

This crawler runs asynchronously over the topology data. The crawler POST response returns an EntityId header, which you can use to check the progress of the crawler. To check the progress include the EntityId header in a **`GET /mgmt_artifacts/{id}`** call. The response shows the status of the asynchronous crawl.

For more information about the topology service Swagger, see Application and topology APIs.

Restore process for ElasticSearch failed

If the restore of a particular data store, such as ElasticSearch, or custom resource fails, complete the following steps before you can attempt the restore again.

1. Optional. If this cluster was previously configured to enable a backup of ElasticSearch, or you need to rerun restore, complete the following steps:

    1. Remove the backup path and Snapshot location configurations from the AutomationBase CR.

    2. Delete the ElasticSearch backup snapshot PVC by running the following command. This deletion is needed since this PCV is replaced with the ElasticSearch backup data and the **`es-restore1`**.

        ```
        oc delete pvc -n <namespace> iaf-system-elasticsearch-es-snap-es-backup-pvc
        ```

    3. Run the following command to delete the snapshot and **`es-restore1`** if it exists:

        ```
        oc delete restore es-restore1 -n velero
        ```

2. Repeat the process to restore the data store or custom resource to reattempt the failed restore.

Restore fails with CouchDB pod in CrashLoopBackOff

This problem can occur if you restore an IBM Cloud Pak for AIOps backup to a different cluster or namespace and the database files in the persistent storage are not readable or writeable by the root user group.

When this issue occurs, the pod **`c-example-couchdbcluster-m`** has a status of *CrashLoopBackOff*, as in the following example.

```
oc get pods | grep couchdb
c-example-couchdbcluster-m-0                                      1/2     CrashLoopBackOff
533 (4m13s ago)   2d5h
c-example-couchdbcluster-m-0-debug                                1/2     Running
0                 2m39s
```

The logs from the database container in the failing pods contain the message **`Could not open file /data/db/_nodes.couch: permission denied`**, as in the following example:

```
oc logs c-example-couchdbcluster-m-0 -c db --tail=100
[error] 2022-11-10T10:39:34.788855Z couchdb@c-example-couchdbcluster-m-0.c-example-
couchdbcluster-m <0.322.0> -------- Could not open file /data/db/_nodes.couch: permission
denied
```

To resolve this issue, grant the root group write permissions to the folders in the persistent storage that contain the database files.

1. Get the name of the persistent volume that contains the CouchDB database files.

    ```
    oc get pv | grep couch
    ```

    Example output:

    ```
    oc get pv | grep couch
    pvc-d6599cfb-aa38-47be-857e-a2efa9ee0407   5Gi        RWO            Delete
    Bound    katamari/data-c-example-couchdbcluster-m-0                           rook-
    cephfs              2d5h
    ```

2. Find the worker node for the failing CouchDB pod.

```
oc get pod -o wide c-example-couchdbcluster-m-0
```

Example output:

```
oc get pod -o wide c-example-couchdbcluster-m-0
NAME                            READY   STATUS             RESTARTS        AGE    IP
NODE                                      NOMINATED NODE   READINESS GATES
c-example-couchdbcluster-m-0   1/2     CrashLoopBackOff   537 (76s ago)   2d5h
10.254.14.54   worker1.bcdr-test-12345678.mysite.com   <none>           <none>
```

3. Debug the worker node that you identified in the previous step.

```
oc debug node/<worker_node>
```

Where **<worker_node>** is the value returned for *NODE* in the previous step.

Example output:

```
oc debug node/worker1.bcdr-test-12345678.mysite.com
Starting pod/worker1bcdr-rtp-26101516cpmyclustercom-debug ...
To use host binaries, run `chroot /host`
Pod IP: 10.22.1.1
If you don't see a command prompt, try pressing enter.
```

4. Find the mount point of the persistent volume.

```
mount | grep <pv>
```

Where **<pv>** is the persistent volume name returned in step 1.

Example output:

```
sh-4.4# mount | grep pvc-d6599cfb-aa38-47be-857e-a2efa9ee0407
172.30.2.127:6789,172.30.43.40:6789,172.30.11.17:6789:/volumes/csi/csi-vol-0bb3b8e3-5f26-
11ed-97a6-0a580afe2807/af21d8e8-f29e-4111-8fcd-c4366d604c99 on
/host/var/lib/kubelet/plugins/kubernetes.io/csi/pv/pvc-d6599cfb-aa38-47be-857e-
a2efa9ee0407/globalmount type ceph (rw,relatime,seclabel,name=csi-cephfs-node,secret=
<hidden>,acl,mds_namespace=rook-cephfilesystem)
172.30.2.127:6789,172.30.43.40:6789,172.30.11.17:6789:/volumes/csi/csi-vol-0bb3b8e3-5f26-
11ed-97a6-0a580afe2807/af21d8e8-f29e-4111-8fcd-c4366d604c99 on
/host/var/lib/kubelet/pods/26fc2b39-aaf0-4add-8649-
6cfd56fc4a3b/volumes/kubernetes.io~csi/pvc-d6599cfb-aa38-47be-857e-a2efa9ee0407/mount
type ceph (rw,relatime,seclabel,name=csi-cephfs-node,secret=
<hidden>,acl,mds_namespace=rook-cephfilesystem)
```

The mount point is the directory ending with **mount**. In the above example output this is
**/host/var/lib/kubelet/pods/26fc2b39-aaf0-4add-8649-
6cfd56fc4a3b/volumes/kubernetes.io~csi/pvc-d6599cfb-aa38-47be-857e-a2efa9ee0407/mount**

5. Change the permissions on the mount directory by changing to the directory and running **chmod**.

```
cd <mount_dir>
chmod -R g+w
ls -lah db/
exit
```

Where **<mount_dir>** is the directory you identified in the previous step.

Example output:

```
sh-4.4# cd /host/var/lib/kubelet/pods/26fc2b39-aaf0-4add-8649-
6cfd56fc4a3b/volumes/kubernetes.io~csi/pvc-d6599cfb-aa38-47be-857e-a2efa9ee0407/mount
sh-4.4# ls
db  view_index
sh-4.4# ls -lah db/
total 37K
drwxr-xr-x. 4 1000650000 root    4 Nov  7 04:55 .
drwxrwxrwx. 5 root       root    3 Nov 10 10:52 ..
drwxr-xr-x. 2 1000650000 root    0 Nov  7 04:54 .delete
-rw-r--r--. 1 1000650000 root  29K Nov  7 05:31 _dbs.couch
-rw-r--r--. 1 1000650000 root 8.3K Nov  7 04:54 _nodes.couch
drwxr-xr-x. 4 1000650000 root    2 Nov  7 04:55 shards
```

```
sh-4.4# chmod -R g+w .
sh-4.4# ls -lah db/
total 37K
drwxrwxr-x. 4 1000650000 root    4 Nov  7 04:55 .
drwxrwxrwx. 5 root       root    3 Nov 10 10:52 ..
drwxrwxr-x. 2 1000650000 root    0 Nov  7 04:54 .delete
-rw-rw-r--. 1 1000650000 root  29K Nov  7 05:31 _dbs.couch
-rw-rw-r--. 1 1000650000 root 8.3K Nov  7 04:54 _nodes.couch
drwxrwxr-x. 4 1000650000 root    2 Nov  7 04:55 shards
sh-4.4# exit
```

6. Run the following command to verify that the CouchDB pod has a status of **`Running`**.

   ```
   oc get pods | grep couchdb
   ```

   Example output:

   ```
   c-example-couchdbcluster-m-0                                      2/2     Running
   539 (87s ago)     2d5h
   ```

LDAP user login is not working after a restore

Follow the steps to solve the problem:

1. Log in to the console as the default admin user.
2. From the main navigation menu, click **Administer** > **Identify and access**.
3. Select the LDAP connection, and click **Edit connection**. Edit the LDAP connection with the correct information.
4. Click **Test connection**.
5. Click **Save** once the connection is success.
6. log in to the console with the LDAP user's credentials.

Data is not being processed after restoration

After you complete a restore, you might notice that data for some AI modeling features, such as log anomaly detection, change risk, similar tickets, might not be processed.

This can occur when expected Kafka topics for the AI modeling is not present after the restore completes. To resolve this problem, restart the **`aimanager-aio-controller`** pod. This restart can result in the creation of the expected Kafka topics.

Applications still show non-existent active incidents and alerts notations

After you complete a restore, you might notice notice that active incidents and alerts notations display for applications when the incidents and notations no longer exist. When the topology data store is backed up and restored, it contains restored data which is out of sync with the rest of the system due to the way the data is modelled against groups, applications, and resources within the topology.

If your data is outdated, you need to manually delete the topology association between the incidents and applications. Use the topology API to clear out any groups or events. Once removed, the changes are incorporated into the topology view.

To remove groups (incident groups of entity **`waiopsStory`**) use the API:

- **`GET /topology/groups?_type=waiopsStory`**
- **`DELETE /topology/groups?_type=waiopsStory`**

To remove events, remove the incorrect status from the topology with the following API:

- **`GET /topology/status`**
- **`DELETE /topology/status`**

Alternatively, you can use the *Resource management* tool in the UI to remove non-existant alert indiciations from a topology. When you are viewing a topology in the UI, open the **Settings** menu and select **Topology configuration**. From the**Data administration routines** page, run the **Status clear** routine to remove the alert indications.

Topology observer jobs do not run after restore

After running the restore procedure, some observer jobs are offline, have a **Status** of **`Error`**, or have a **Status** of **`Running`** even though job runs are being skipped.

For jobs with a job type of **`Repeating Schedule`**, this problem can occur when a restore ran from a backup that was taken while the observer job was waiting for data processing to finish. After the restore completes, the observer then starts with the job still in a FINISHING state. The observer waits for its job to finish, but the job cannot finish because the referenced data was not backed up, and so the job becomes stuck. Messages similar to the following might be seen in the observer log:

```
execution skipped because current job  had state: FINISHING
```

For other jobs that do not have a job type of **`Repeating Schedule`**, the observer log shows that it is looking for a vertex that no longer exists:

```
WARN   [2022-09-13 16:06:24,265] [pool-5-thread-1] c.i.i.t.o.t.ObserverVertex -  Failed to
poll observer vertex cp4waiops-cartridge.github-observer.  Response :
InboundJaxrsResponse{context=ClientResponse{method=POST, uri=https://aiops-topology-
topology:8080/1.0/topology/mgmt_artifacts/MKSrcRpkRC6WmWVGGdSukA, status=404, reason=Not
Found}}
```

To resolve these problems, complete the following steps:

1. Locate jobs that are stuck in the **`FINISHING`** state.

   For jobs with a **Job type** of **`Repeating schedule`**, check if the job is stuck by waiting for the normal job duration or by checking the job history. If the job has been stuck for a long time, subsequent runs will be skipped. Use the following steps to navigate to the Observer configuration user interface (UI), view the observer job history and look for job runs that have been continuously skipped since the restore completed.

   1. Log in to IBM Cloud Pak Automation console.
   2. Expand the navigation menu (four horizontal bars), then click **Define** > **Data and tool connections**.
   3. On the *Data and tool connections* page, click **Add connection**.
   4. On the *Add connections** page, click **Topology** in the *Category* list that is next to the list of all connections.
   5. Click the **configure, schedule, and manage other observer jobs** link in the description for the topology connections.
   6. Click the three dots icon at the end of the line for the observer job, and select **View history**.
   7. Click **All**, and look for runs with **`Skipped`** status and a **Details** entry of **`execution skipped because current job had state: FINISHING`**.

   For more information on viewing observer jobs, see the section **`To access the Observer Configuration UI`** in Defining observer jobs for application and topology data.

2. Attempt to restart the observer job.

   1. For each observer job with a **Job type** of **`Repeating schedule`** that you identified in FINISHING state, edit the job in the Observer configuration UI by nominally changing the **Optional description** field, and saving it.

   2. For other observer jobs that do not have a **Job type** of **`Repeating schedule`** and have a missing vertex issue, delete the related observer pod.

      ```
      oc delete pod <pod_name> -n <namespace>
      ```

      Where

      - **`<pod_name>`** is the observer pod to be deleted.
      - **`<namespace>`** is the project (namespace) where IBM Cloud Pak for AIOps is installed.

      For example, for an Instana observer job that has a missing vertex issue, delete the Instana observer pod:

      ```
      oc get pods -n | grep instana
      aiopsedge-instana-topology-integrator-6b777bd6b4-6nvqx 1/1 Running 0 8d

      oc delete pod aiopsedge-instana-topology-integrator-6b777bd6b4-6nvqx -n <cp4aiops-
      namespace>
      ```

      This causes the job to stop and restart, and to be shown as **Running**, **Scheduled**, or **Ready**.

      A message similar to the following is seen in the observer log:

      ```
      WARN   [2022-08-22 16:19:34,706] [pool-10-thread-1] c.i.i.t.o.a.JobManager -  stop -
      Stopped job for tenantId:cfd95b7e-3bc7-4006-a4a8-a73a79c71255, uniqueId:1100
      INFO   [2022-08-22 16:19:34,712] [cfd95b7e-3bc7-4006-a4a8-a73a79c71255:1100]
      c.i.i.t.o.f.j.LoadFileJob -  cfd95b7e-3bc7-4006-a4a8-a73a79c71255:5eb27ee0-772c-
      ```

```
4d65-bc0f-4e56b6b7913b Interrupted while waiting for state JobState [state=FINISHED,
reason=Job finished after restart, date=Mon Aug 22 16:09:45 GMT 2022]
INFO   [2022-08-22 16:19:34,944] [cfd95b7e-3bc7-4006-a4a8-a73a79c71255:1100]
c.i.i.t.o.f.j.LoadFileJob -  cfd95b7e-3bc7-4006-a4a8-a73a79c71255:1100 Observer job
finished with state STOPPED
```

Some jobs might still be shown with **Error** if an attempt was made to run the stuck job before the previous remediating steps were attempted. This causes the observer to fail the job, as a job already exists with the same name. In this scenario, a warning similar to the following is seen in the observer log:

```
WARN   [2022-08-23 15:58:50,431] [pool-10-thread-1] c.i.i.t.o.a.o.RESTApiException -
pool-10-thread-1 - APIMessage: { httpCode=422, _error={ message=Job Creation Failure,
level=warning, description=Cannot create observation job, causes=[{ message=Duplicate
unique id, level=error, description=A job with the same unique id '1100' is already being
processed., field=unique_id }] } }
```

If there are no longer any observer jobs with a **Status** of **Error**, then exit this troubleshooting.

3. Complete the following steps only if one or more of your observer jobs are still shown with a **Status** of **Error**.

    1. Find the name of the topology pod.

        ```
        oc get pods | grep topology-topology
        ```

    2. Remote shell into the topology container.

        ```
        oc rsh <topology_pod> -c <release_name>-topology-topology
        ```

        Where

        - **<topology_pod>** is the pod that was returned in the previous step.
        - **<release_name>** is the name of your IBM Cloud Pak for AIOps instance.

4. Find observers that still have a job that is stuck in the **FINISHING** state.

    Run the following command, and note down the returned **observerName** and **_id**.

    ```
    curl -X GET "https://localhost:8080/1.0/topology/mgmt_artifacts?
    _filter=hasState%3DFINISHING&_field=keyIndexName&_field=hasState&_field=observerName&_typ
    e=ASM_OBSERVER_JOB" -H  "accept: application/json" -H  "X-TenantID: cfd95b7e-3bc7-4006-
    a4a8-a73a79c71255" --insecure -u ${ASM_USER}:${ASM_PASS} | jq
    ```

    Note: ASM_USER and ASM_PASS are environment variables available within the topology container.

    Example output:

    ```
    {
     "_executionTime": 13,
     "_offset": 0,
     "_limit": 50,
     "_items": [
       {
         "keyIndexName": "dns-observer:bbc",
         "_id": "z7ovGeNrSKSpMlWY8n7Mfw",
         "observerName": "dns-observer",
         "hasState": "FINISHING"
       }
     ]
    }
    ```

5. Run the following steps for each observer job returned by the previous step.

    1. Change the state of the job by running the following command.

        ```
        curl -X POST "https://localhost:8080/1.0/topology/mgmt_artifacts/<id>" -H  "accept:
        application/json" -H  "X-TenantID: cfd95b7e-3bc7-4006-a4a8-a73a79c71255" -H
        "Content-Type: application/json" -d "{  \"hasState\": \"FINISHED\"}" --insecure -u
        ${ASM_USER}:${ASM_PASS}
        ```

        Where **<id>** is the **_id** of the stuck job that you identified in step 4.

    2. Find the pods for the observer that you need to restart.

```
oc get pods -n <namespace> | grep <stuck_observer_name>
```

Where

- **<stuck_observer_name>** is the name of the stuck observer that you identified in step 4.
- **<namespace>** is the project (namespace) where IBM Cloud Pak for AIOps is installed.

3. Delete the pods for the observer.

```
oc delete po <observername_pods> -n <namespace>
```

Where

- **<observername_pods>** are the pods returned in the previous step.
- **<namespace>** is the project (namespace) where IBM Cloud Pak for AIOps is installed.

4. Rerun the observer job.

On the observer configuration UI, click **Run** on any job for the observer that is in the **Error** state. The observer job now runs successfully, and has a status of **Scheduled** or **Ready**.

Restore process terminated mid-process with partial data available

If the restore process does not complete as expected for a data store, such as if it is aborted or terminated during its run, data might not be restored correctly. This incomplete restored data neeeds to be removed before you run the restore process again. To remove the data, you need to run a post-restore cleanup script to clean up the data store.

To run a script, complete the following steps:

1. Define the following environment variable on your workstation:

```
export WORKDIR="<Path>/bcdr/4.2.0/"
```

Where **<Path>** is the path to where you downloaded and extracted the IBM Cloud Pak® for AIOps backup and restore files.

2. Change to the **restore** directory where the post-restore cleanup script is located:

```
cd <Path>/bcdr/4.2.0/restore/<data_store>/
```

Where

- **<Path>** is the path to where you downloaded and extracted the IBM Cloud Pak® for AIOps backup and restore files.
- **<data_store>** is the directory for the data store or custom resource that needs to be cleaned up. For example, **couchdb** is the directory for CouchDb.

3. Run the post-restore cleanup script:

```
nohup ./<data-store>-post-restore.sh.sh > <data-store>-post-restore.log &
```

Where **<data_store>** is the data store or custom resource that needs to be cleaned up.

4. Run the restore process for that data store or resource again.

For example, if the restore job for restoring the Cassandra data store aborted, you need to run the **cassandra-post-restore.sh** post-restore script that is stored in the **bcdr/restore/cassandra** directory to clean the data. Then, run the **cassandra-native-post-restore.sh** post-restore script.

The following table lists the cleanup script to run for each data store:

Table. Cleanup script for components

Component or Database	Cleanup script
Cassandra	Run first: **bcdr/4.2.0/restore/cassandra/cassandra-post-restore.sh** Then, run: **bcdr/4.2.0/restore/cassandra/cassandra-native-post-restore.sh**
CouchDB	**bcdr/4.2.0/restore/couchdb/couchdb-post-restore.sh**
Elasticsearch	**bcdr/4.2.0/restore/elasticsearch/es-post-restore.sh**
Metastore	**bcdr/4.2.0/restore/metastore/metastore-post-restore.sh**

Component or Database	Cleanup script
`Minio`	`bcdr/4.2.0/restore/minio/minio-post-restore.sh`
`Postgres`	`bcdr/4.2.0/restore/postgres/postgres-post-restore.sh`
IBM Cloud Pak foundational services	`bcdr/4.2.0/restore/common-services/cs-post-restore.sh`
Connection CR	N/A
Secure Tunnel CR	`bcdr/4.2.0/restore/other-resources/tunnel-cr-post-restore.sh`

Velero restore is stuck in an `In progress` state

If you notice that the Velero pod is stuck in an `In progress` state, complete the following steps to stop the restore process:

1. Delete the velero pod by running the following command:

   ```
   oc delete pod <velero pod name> -n <velero installed namespace>
   ```

2. Delete the restore that is stuck in progress by running the following command:

   ```
   velero delete restore <restore name>
   ```

   Wait for the completion of the restore job.

Helm install backup job command failed

When you are running the `helm install restore-job clusterrestore-0.1.0.tgz` command, you might encounter the command failing with an error that is similar to the following error:

```
Error: admission webhook "trust.hooks.securityenforcement.admission.cloud.ibm.com" denied the
request:
Deny "<Image Registry Server URL>/<Repository>/cp4waiops-bcdr:latest", no matching
repositories in ClusterImagePolicy and no ImagePolicies in the "velero" namespace
```

Where:

- `<Image Registry Server URL>` is the image registry server URL.
- `<Repository>` is the repository where you put the image.

If you encounter this error, complete the following steps to resolve the issue:

1. Uninstall the `restore-job` job by running the following command:

   ```
   helm uninstall restore-job -n velero
   ```

2. Create a `restore-image-policy.yaml` file and add the following content within the file:

   ```
   apiVersion: securityenforcement.admission.cloud.ibm.com/v1beta1
   kind: ClusterImagePolicy
   metadata:
     name: restore-image-policy
   spec:
    repositories:
     - name: "<Image Registry Server URL>/<Repository>/cp4mcm-bcdr:latest"
       policy:
   ```

3. Apply the policy by running the following command:

   ```
   oc apply -f restore-image-policy.yaml
   ```

4. Deploy the backup job by running the following command:

   ```
   helm install restore-job clusterbackup-0.1.0.tgz
   ```

Cassandra restore fails with `org.apache.cassandra.io.FSReadError` exception

Policies are not restored after restore completed. Although there were a couple of key repair failures, the log indicated the Cassandra restore completed. However, the policies from the backup cluster were not restored.

You can encounter an issue where the Cassandra restore fails. If this issue occurs, the **aiops-topology-cassandra-2** pod is in **0/1** status and the pod log contains an error message that is similar to the following error message:

```
ERROR [CompactionExecutor:8] 2022-11-29 21:39:17,324 CassandraDaemon.java:244 - Exception in
thread Thread[CompactionExecutor:8,1,main]
org.apache.cassandra.io.FSReadError: java.io.IOException: Channel not open for writing -
cannot extend file to required size
```

If this error occurs, complete the following steps for the Cassandra statefulset:

1. Scale down the statefulset **aiops-topology-cassandra** to **0**.

   Important: Make a note of the current scaling of the Cassandra statefulset before you scale down.

   ```
   oc scale statefulsets aiops-topology-cassandra --replicas 0 -n <namespace>
   ```

   Where: **<namespace>** is the namespace where IBM Cloud Pak for AIOps is installed.

2. Increase the statefulset memory limit to 32 GB.

   1. Access the resource for editing:

      ```
      oc edit statefulset aiops-topology-cassandra -n <namespace>
      ```

   2. Increase the memory limit:

3. Scale the statefulset back to the initial number of replicas.

   ```
   oc scale statefulsets aiops-topology-cassandra --replicas=<number_of_replicas> -n
   <namespace>
   ```

   Where:

   - **<namespace>** is the namespace where IBM Cloud Pak for AIOps is installed.
   - **<number_of_replicas>** is the number of replicas the StatefulSet it to be scaled up to.

4. Run the restore again.

5. Scale down the statefulset **aiops-topology-cassandra** to **0** again.

   ```
   oc scale statefulsets aiops-topology-cassandra --replicas 0 -n <namespace>
   ```

6. Change the statefulset memory limit to 16 GB .

   1. Access the resource for editing:

      ```
      oc edit statefulset aiops-topology-cassandra -n <namespace>
      ```

   2. Decrease the memory limit to 16 GB.

7. Scale the statefulset back to the initial number of replicas.

   ```
   oc scale statefulsets aiops-topology-cassandra --replicas=<number_of_replicas> -n
   <namespace>
   ```

   Where:

   - **<namespace>** is the namespace where IBM Cloud Pak for AIOps is installed.
   - **<number_of_replicas>** is the number of replicas the StatefulSet it to be scaled up to.

Connections missing after a restore

After you complete a full restore, if you do not see all of the data connections that you expect, run the command to restore data for the **Metastore** component to restore the connections. For instructions, see Restoring individual components.

Netcool connection is missing details after restore

If you had a Netcool connection defined, then the Netcool connector may be missing some details after restore, such as the ObjectServer details. If you try to edit these details then the test connection still fails. To resolve this problem, delete the Netcool connection, and then create a new Netcool connection.

Troubleshooting the Infrastructure Automation restore

If you are also restoring Infrastructure Automation data and encounter an issues with the restore process for Infrastructure Automation or encounter an issue with data not being available or processed after the restore, see Troubleshooting the Infrastructure Automation restore

In addition, review the Known issues with the backup and restore process.

Known issues with the backup and restore process

- Elasticsearch health status yellow after restore

Elasticsearch health status yellow after restore

When you are restoring an Elasticsearch backup to a new single node Elasticsearch cluster, the Elasticsearch database might not work as expected. Instead, the Elasticsearch cluster health shows a yellow status after the restore completes.

Upgrading IBM Cloud Pak for AIOps backup and restore artifacts

Use this procedure if you want to run backup or restore on an upgraded IBM Cloud Pak® for AIOps deployment.

If you run a backup or restore on an IBM Cloud Pak for AIOps deployment, the backup and restore scripts must always be the same version as your deployment. For example, if you run backup on a IBM Cloud Pak for AIOps v4.2.0 deployment, you must use a v4.2.0 backup script.

If you previously set up backup or restore on a deployment that you have now upgraded, then you must clean up any existing backup or restore artifacts. If you then want to run a new backup or restore, you must get the new version of the backup or restore scripts.

Note: If you have never set up backup or restore and you want to run backup or restore, then use the instructions in IBM Cloud Pak for AIOps backup and restore.

Before you begin

Create environment variables for use in the commands in the following steps.

```
export IMAGE_REGISTRY=<image_registry>
export USERNAME=<username>
export PASSWORD=<passowrd>
export EMAIL=<email>
export REPOSITORY=<repository>
export VELERO_NAMESPACE=<velero_ns>
```

Where:

- `<image_registry>` is the image registry server URL.
- `<username>` is the username to log in to the image registry server.
- `<password>` is the password to log in to the image registry server.
- `<email>` is the email for image registry server.
- `<repository>` is the repository where you want to put the images.
- `<velero_ns>` is the namespace that Velero is installed in.

Set up backup on an upgraded IBM Cloud Pak for AIOps deployment

Use the following steps to upgrade your backup artifacts.

1. Uninstall the backup helm chart.

Run the following command in the namespace that **Red Hat OpenShift APIs for Data Protection** (OADP) is installed in.

```
helm uninstall backup-job
```

2. If your image registry server is unchanged since you previously set up backup, then skip this step. Otherwise, delete and create a new image pull secret for backup.

```
oc delete secret backup-secret -n ${VELERO_NAMESPACE}
oc create secret docker-registry backup-secret -n ${VELERO_NAMESPACE} --docker-
server=${IMAGE_REGISTRY} --docker-username=${USERNAME} --docker-password=${PASSWORD} --
docker-email=${EMAIL}
```

3. Clone the IBM Cloud Pak for AIOps samples GitHub repository that holds the backup and restore scripts. Clone this repository by running the following command:

```
git clone https://github.com/IBM/cp4waiops-samples.git
```

4. Build the Docker image.

    1. Change to the directory that you downloaded the backup and recovery files to in the previous step.

    ```
    cd <Path>/bcdr/4.2.0/
    ```

    Where **<Path>** is the path to where you downloaded and extracted the IBM Cloud Pak for AIOps backup and restore files.

    2. Build the **cp4waiops-bcdr** Docker image by running following command:

    ```
    docker build -t cp4waiops-bcdr:v4.2.0 -f Dockerfile .
    ```

5. Tag and push the Docker image to the image registry.

    1. Run the following command to tag your image:

    ```
    docker tag cp4waiops-bcdr:v4.2.0 ${IMAGE_REGISTRY}/${REPOSITORY}/cp4waiops-
    bcdr:v4.2.0
    ```

    2. Run the following command to log in to the image registry:

    ```
    docker login ${IMAGE_REGISTRY} -u ${USERNAME}
    ```

    3. Run the following command to push your image to the image registry:

    ```
    docker push ${IMAGE_REGISTRY}/${REPOSITORY}/cp4waiops-bcdr:v4.2.0
    ```

6. If you need to run backup for IBM Cloud Pak for AIOps v4.2.0, follow the instructions in Backing up IBM Cloud Pak for AIOps data from step *Package the Helm Chart* onwards.

Set up restore on an upgraded IBM Cloud Pak for AIOps deployment

Use the following steps to upgrade your restore artifacts.

1. Uninstall the restore helm chart.

    Run the following command in the namespace that **Red Hat OpenShift APIs for Data Protection** (OADP) is installed in.

    ```
    helm uninstall restore-job
    ```

2. If your image registry server is unchanged since you previously set up restore, then skip this step. Otherwise, delete and create a new image pull secret.

```
oc delete secret restore-secret -n ${VELERO_NAMESPACE}
oc create secret docker-registry restore-secret -n ${VELERO_NAMESPACE} --docker-
server=${IMAGE_REGISTRY} --docker-username=${USERNAME} --docker-password=${PASSWORD} --
docker-email=${EMAIL}
```

3. Setup restore for IBM Cloud Pak for AIOps v4.2.0.

   If you need to run restore for IBM Cloud Pak for AIOps v4.2.0, follow the instructions in Restoring IBM Cloud Pak for AIOps to set up restore for IBM Cloud Pak for AIOps v4.2.0.

   Important:

   - The `cp4waiops-bcdr` docker image must be the same version as your IBM Cloud Pak for AIOps deployment, and must be the same `cp4waiops-bcdr` docker image that you created for setting up backup on your upgraded deployment. Follow these instructions to ensure that your `cp4waiops-bcdr` docker image is v4.2.0 for your IBM Cloud Pak for AIOps v4.2.0 deployment.
   - Skip the step that creates an image pull secret, as the required secret exists already.

Administering CouchDB data storage

Manage disk space used by runbook executions to avoid outages when the available disk space has been exhausted.

You can offload and delete historic runbook executions to reduce database footprint and increase database performance. Offloaded runbook executions are deleted from the production database and stored on a file system instead to preserve them for future audit purposes. They can be stored either in JSON or HTML format. A script for offloading runbook executions is included in the product and can be run by using the fully automated sample runbook **Example: Automatically reduce amount of data in runbook data stores**. Running this fully automated runbook has the negative side effect that the API key (which is contained in the parameters of the fully automated runbook) is stored and can be viewed within the runbook execution details. This description shows how you can avoid this limitation and still use the script that is available at the core of the fully automated runbook.

Runbook executions can be offloaded to store them somewhere else, in a different format (JSON or HTML), or to reduce the amount of data stored in the database. This is especially important if either the database is getting short of disk space, or a performance degradation that is caused by too much data is becoming noticeable. It is also a good practice to regularly offload runbook executions that exceed a certain age and are therefore no longer required on the production system.

When you offload and delete runbook execution records, the referenced action execution records are included in the offloading output and deleted as well. However, additional action execution records might remain for runbook executions that were already deleted separately, or were executed outside of a runbook context. You can use the following instructions on how to offload and delete those action execution records. Make sure to delete the action execution records only after the runbook execution records for the same time frame have been offloaded and deleted. Otherwise, the runbook execution records contain broken links to action execution records that have been deleted already.

Note:

- Once runbook executions are offloaded and deleted, they are no longer accessible from the runbook history page and are not reflected in the aggregated statistics of their runbook anymore.
- The size of the database does not shrink immediately after the runbook executions have been deleted, but only after the next CouchDB database compaction has been performed by CouchDB itself.

It is recommended to set up a scheduled task that offloads runbook execution data is regularly. For your planning you should consider the following aspects:

- What is the maximum size of the persistent volume that hosts the database where the runbook execution data is stored? For example, 20 GB.
- What is the average size of a single runbook execution in your environment? For simple runbooks, you can assume a value of about 15 KB per runbook execution.
- How many runbooks are executed per day?

Create an initial schedule for offloading the runbooks based on these values, allowing some buffer for days with higher load. Observe the behavior for some time, and tune the frequency of the scheduled task or the parameters of the runbook to match your needs.

The offloading script must run on a Linux system that meets the following requirements:

- Network access to the OpenShift cluster where IBM Cloud Pak® for AIOps is installed.

- The `curl` tool must be installed.
- The `jq` tool must be installed. Starting with RHEL 8.0, `jq` can be installed by using the AppStream repository. On earlier RHEL versions, `jq` can be installed by following the instructions on https://stedolan.github.io/jq/.
- Some free disk space must exist to store copies of the offloaded runbook executions. See the FOLDER parameter for more information.
- The `base64` encoding tool must be installed.

The offloading script requires several environment variables to be set:

- API: The Runbook API URL as described in Automation (runbook) API. For example, set the value of the API parameter to `https://cpd-cp4waiops.apps.mycomp.com/aiops/api/story-manager/rba/v1`.
- APIKEY_NAME: The user ID that you used to log in when creating an API key. As described in Accessing APIs, section *Get access token by using your API key*, step *Generate your platform API key*.
- APIKEY_PW: The API key password that you created when following the instructions in Accessing APIs, section *Get access token by using your API key*, step *Generate your platform API key*.
- FOLDER: The fully qualified path name of an existing directory on the Linux system that will be used for storing the offloaded runbooks.
- MODE: Used to specify a format for the offloaded runbook executions. You can choose between `html` and `json`.
- DELETE: Whether to delete the offloaded runbook executions from the runbook database after successful offload. Recommended to be set to `false` while performing early tests, but set to `true` once you are ready to delete documents and reduce the size of the database.
- OLDER_THAN_MONTHS: A number of months to specify the timeframe for offloading runbook executions. Only runbook executions older than the specified number of months are offloaded. If you set this value to `0` then all runbook executions that were created before today will be offloaded. You can set this parameter to `-1` and instead use the parameters AFTER and BEFORE if you need to have more granular control on the time period.
- AFTER: Only offload execution records that are started after this timestamp. You can specify a date like `2022-03-31` or a timestamp in the moment.js format, for example, `2022-03-31T12:59:59`. If you want to use this parameter, the parameter OLDER_THAN_MONTHS must be set to the empty string.
- BEFORE: Only offload execution records that are started before this timestamp. You can specify a date like `2022-04-04` or a timestamp in the moment.js format, for example, `2022-04-04T12:59:59`. If you want to use this parameter, the parameter OLDER_THAN_MONTHS must be set to the empty string.

1. One-time set up to create the script

1. Create an API key as described in Accessing APIs, section *Get access token by using your API key*, step *Generate your platform API key*.
2. If it does not exist already, create the SSH integration as described in Creating SSH integrations. You do not need to define a jumpserver and you do not need to copy the SSH key to any target system for the purpose of this procedure.
3. Load the sample runbooks as described in Load and reset example runbooks.
4. Switch to the **Actions** tab.
5. Select the action **Example: Reduce amount of data in runbook data stores** and click **Edit**.
6. Copy the content of the **Script** field to the clipboard.
7. Log in to the Linux system and change to the directory where you want the script to be stored.
8. Create a file called "offloadRunbookExecutions.sh" and paste the content of the clipboard into this file.
9. Make the file "offloadRunbookExecutions.sh" executable, for example, run the following command:

   ```
   chmod a+x offloadRunbookExecutions.sh
   ```

10. Create a file called "offload.config" and paste the following content into this file:

    ```
    export API="https://cpd-cp4waiops.apps.mycomp.com/aiops/api/story-manager/rba/v1"
    export APIKEY_NAME="<api_key_user_id>"
    export APIKEY_PW="<api_key_password>"
    export FOLDER=$(pwd)
    export MODE=json
    export DELETE=false
    ```

```
export OLDER_THAN_MONTHS=-1
export AFTER="2022-03-01"
export BEFORE="2022-04-13T12:00:00"
#export CURL_OPTIONS="-k"
```

11. Adjust the values of the environment variables as needed (see the variable descriptions listed earlier) and save the file.

2. Test the script

1. Run the following command to load the environment variables into the current shell environment:

   ```
   . ./offload.config
   ```

2. Run the script:

   ```
   ./offloadRunbookExecutions.sh
   ```

3. If the action fails and you see the error **`curl: \(60\) SSL certificate problem: self-signed certificate in certificate chain`**, you can either replace the IBM Netcool Operations Insight server certificate with a trusted certificate, or enable the environment variable **`CURL\_OPTIONS="-k"`** to ignore the certificate errors.

4. If the action fails and you see the following error:

   ```
   ARBD0420E: Could not fetch list of runbook instances:
   curl: (22) The requested URL returned error: 500 Internal Server Error
   ```

   Then reduce the length of the time interval for offloading execution records (for example, by increasing the value of OLDER_THAN_MONTHS) and try again.

5. Review the results that were created by the script. If needed adjust the environment variables and repeat the procedure until you are fine with the results.

3. Schedule the script

1. Convert this script command to the format that is expected by your scheduling tool, and set up a schedule to regularly offload the runbook data.

2. Make sure that the DELETE environment variable is set to true once you are finished with the testing of the script and the schedule, and when you are ready to delete the offloaded runbook execution instances from the database.

Offloading and deleting remaining action execution records

Once you have deleted the runbook execution records for a particular time frame you should also check if there are remaining action execution records (also called action instances) for that time frame, and delete them as well.

Step 1: Prepare and verify the API URL and the credentials Adjust the values of the environment variables as needed (see the variable descriptions listed earlier) and run the following commands:

```
export API="https://cpd-cp4waiops.apps.mycomp.com/aiops/api/story-manager/rba/v1"
export APIKEY_NAME="<api_key_user_id>"
export APIKEY_PW="<api_key_password>"
export CURL_AUTHENTICATION=$(echo -n "${APIKEY_NAME}:${APIKEY_PW}" | base64 -w 0 | echo
"Authorization: ZenApiKey $(cat -)")
export CURL_OPTIONS="-ks"
curl "${CURL_OPTIONS}" -H "${CURL_AUTHENTICATION}" "${API}/status"
```

The command should run without any authentication errors, and the output should look similar to this (for the purpose of this task, you can ignore the value of the "connection" property, in particular any potential "errors" and "warnings" within the respective properties):

```
{"rbs":0,"as":0,"db":0,"connection":{"AWX":0,"IMPACT":2,"SCRIPT":0,"GITHUB":2},"errors":
[],"warnings":[]}
```

Step 2: Get the remaining action execution records Adjust the "to" time stamp in the following command and run it to retrieve a list of the records that were created up to that date, and store their IDs in an environment variable:

```
curl "${CURL_OPTIONS}" -H "${CURL_AUTHENTICATION}" "${API}/actioninstances?from=2010-01-
01&to=2022-12-01" > actionInstances.json
export ACTION_INST_IDS=`cat actionInstances.json | jq '.[]' | jq -r '.[]._actionInstanceId'`
```

The records in `actionInstances.json` do not contain the actual output of each action. If you want to preserve the output for each of the actions, run the following command:

```
for actionInstId in ${ACTION_INST_IDS}; do echo "Getting details for ${actionInstId}";
curlOutput=$(curl -X GET "${CURL_OPTIONS}" -H "${CURL_AUTHENTICATION}"
"${API}/actioninstances/${actionInstId}" > ${actionInstId}.json); if (test ${#curlOutput} -gt
0); then echo ${curlOutput}; fi; done
```

Store the each of these action execution record files as appropriate.

Step 3: Delete the remaining action execution records Run the following command to delete the records:

```
for actionInstId in ${ACTION_INST_IDS}; do echo "Deleting ${actionInstId}"; curlOutput=$(curl
-X DELETE "${CURL_OPTIONS}" -H "${CURL_AUTHENTICATION}"
"${API}/actioninstances/${actionInstId}"); if (test ${#curlOutput} -gt 0); then echo
${curlOutput}; fi; done
```

Updating usage data collection preferences

To assist in the development of IBM Cloud Pak for AIOps, daily aggregated usage data is collected to analyse how IBM Cloud Pak for AIOps is used. The collection of usage data is enabled or disabled when you install or upgrade to IBM Cloud Pak for AIOps 4.2.0, but can be changed at any time.

This usage data is collected by the `cp4waiops-metricsprocessor` pod, and is sent to and stored in IBM controlled GDPR-compliant systems. The collection of usage data is enabled by default, but can be disabled. For transparency, the `cp4waiops-metricsprocessor` pod's logs will contain all of the information that is collected. The usage data that is collected is all numeric, and does not include email addresses, passwords, or specific details. Only the following data is collected:

- Current number of applications
- Current number of alerts (all severities aggregated)
- Current number of incidents (all priorities aggregated)
- Current number of policies (includes predefined and user created)
- Current number of runbooks executed since install
- Current number of connectors of each type (For example Service Now, Instana, Falcon Logscale)

Disabling the collection of usage data

1. Set environment variables.

   ```
   export CUSTOMER_NAME=<your company name>
   export CUSTOMER_ICN=<your IBM Customer Number>
   export CUSTOMER_ENVIRONMENT=<Set to `trial`, `poc`, or `production`>
   export PROJECT_CP4AIOPS=<the project where IBM Cloud Pak for AIOps is deployed, usually
   cp4aiops>
   ```

2. Disable usage data collection.

   ```
   oc create secret generic aiops-metrics-processor -n ${PROJECT_CP4AIOPS} --from-
   literal=customerName=${CUSTOMER_NAME} --from-literal=customerICN=${CUSTOMER_ICN} --from-
   literal=environment=${CUSTOMER_ENVIRONMENT} --from-literal=enableCollection=false
   ```

3. Find the name of the metricsprocessor pod by running the following command:

   ```
   oc get pods | grep cp4waiops-metricsprocessor -n ${PROJECT_CP4AIOPS}
   ```

   Example output:

   ```
   cp4waiops-metricsprocessor-f4dc776d-7pwlm                          1/1     Running     0
   19h
   ```

4. Restart the metricsprocessor pod by running the following command:

   ```
   oc delete pod <metrics_processor_pod> -n ${PROJECT_CP4AIOPS}
   ```

   Where **<metrics_processor_pod>** is the pod returned by the previous step.

Enabling the collection of usage data

1. Set environment variables.

   ```
   export CUSTOMER_NAME=<your company name>
   export CUSTOMER_ICN=<your IBM Customer Number>
   export CUSTOMER_ENVIRONMENT=<Set to `trial`, `poc`, or `production`>
   ```

2. Configure the usage data with your customer details.

   ```
   oc create secret generic aiops-metrics-processor -n ${PROJECT_CP4AIOPS} --from-
   literal=customerName=${CUSTOMER_NAME} --from-literal=customerICN=${CUSTOMER_ICN} --from-
   literal=environment=${CUSTOMER_ENVIRONMENT}
   ```

3. If you have a firewall enabled, ensure that outbound traffic to https://api.segment.io is allowed.

4. Find the name of the metricsprocessor pod by running the following command:

   ```
   oc get pods | grep cp4waiops-metricsprocessor -n ${PROJECT_CP4AIOPS}
   ```

   Example output:

   ```
   cp4waiops-metricsprocessor-f4dc776d-7pwlm                          1/1     Running     0
   19h
   ```

5. Restart the metricsprocessor pod by running the following command:

   ```
   oc delete pod <metricsprocessor_pod> -n ${PROJECT_CP4AIOPS}
   ```

   Where **<metricsprocessor_pod>** is the pod returned by the previous step.

Clearing out topology data without reinstalling

You can clear all topology data without having to reinstall your system.

Before you begin

Ensure you have all the required administrator or management permissions to perform administration tasks.

When clearing out the inventory data, you will be prompted for the **user password**. You can retrieve this by using a command as in the following example:

```
oc get secret $release-topology-postgres-user -o jsonpath='{.data.password}' | base64 -d
```

Where **$release** is the release prefix for the topology service, for example **aiops-**.

The values for the **schema**, **database** and **user** are the defaults. You can verify (and adjust if necessary) these values by checking the output of the following commands:

For schema:

```
oc get asmformation $release-topology -o jsonpath='{.spec.helmValues.global.postgres.schema}'
```

For database name:

```
oc get asmformation $release-topology -o jsonpath='{.spec.helmValues.global.postgres.dbname}'
```

For user name:

```
oc get secret $release-topology-postgres-user -o jsonpath='{.data.username}' | base64 -d
```

About this task

To clear all topology data without having to reinstall your system, you perform the following sequence of steps:

1. Stop all observer jobs from running to ensure no new data is received until the old data has been removed.
2. Back up any configurations if required.
3. Set the environment variables manually. This will help with the pre- and post-validation of the topology and inventory databases.
4. Pre-validate the topology and inventory databases to confirm that the data to be removed is present.
5. Run the script to clean the topology and inventory databases.
6. Post-validate to confirm that the data has been removed from the topology and inventory databases.
7. Scale up the topology pods to restart the system and any configured observer jobs.

To clear topology data

1. Stop all observer jobs.

   a. Expand **Define**, click **{{site.data.keyword.ui_data_integrations}}**, click the **Manage observer jobs** tab, then click **Configure, schedule, and manage observer jobs**. The Observer jobs page is displayed.

   b. Ensure that no observer jobs are running or are scheduled to run.

2. Back up IBM Cloud Pak for AIOps configurations if required. For more information, see Backing up and restoring UI topology configuration data in the IBM Netcool Operations Insight documentation.

3. Set the required environment variable.

   a. Set the release and namespace as required

   ```
   export release=aiops
   export NAMESPACE=cp4aiops
   ```

   b. Set the following variables:

   ```
   export POSTGRES_PASS=$(oc get secret $release-topology-postgres-user -o
   jsonpath='{.data.password}' | base64 -d)
   export POSTGRES_USER=$(oc get secret $release-topology-postgres-user -o
   jsonpath='{.data.username}' | base64 -d)
   export CASSANDRA_PASS=$(oc get secret $release-topology-cassandra-auth-secret -o
   jsonpath='{.data.password}' | base64 -d)
   export CASSANDRA_USER=$(oc get secret $release-topology-cassandra-auth-secret -o
   jsonpath='{.data.username}' | base64 -d)
   export POSTGRES_POD=$(oc get clusters.postgresql.k8s.enterprisedb.io $installation-edb-
   postgres --namespace $NAMESPACE -o jsonpath="{.status['currentPrimary']}")
   ```

4. Pre-validate the topology and inventory databases to confirm the presence of the data to be cleared.

   a. Use the following command to verify that the topology database has the janusgraph keyspace:

   ```
   oc exec $release-topology-cassandra-0 -- bash -c "echo 'SELECT * FROM
   system_schema.keyspaces;exit' |cqlsh --ssl -u '$CASSANDRA_USER' -p '$CASSANDRA_PASS'"
   ```

   The following example output indicates that the janusgraph keyspace is present in the topology (Cassandra) database:

   ```
   | keyspace_name | durable_writes | replication
   |
   | ------------- | -------------- | ----------------------------------------------------
   ------------------------------ |
   |         aiops |           True | {'class':
   'org.apache.cassandra.locator.NetworkTopologyStrategy', 'datacenter1': '1'} |
   |        x_in_y |           True | {'class':
   'org.apache.cassandra.locator.NetworkTopologyStrategy', 'datacenter1': '1'} |
   |    janusgraph |           True |   {'class':
   'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'} |
   |   system_auth |           True |   {'class':
   'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'} |
   ```

b. Use the following command to verify that the inventory database contains data (under types, edges and other tables).

```
oc exec -it $POSTGRES_POD -- /bin/bash -c 'export PGPASSWORD='$POSTGRES_PASS'; psql --
host localhost --username '$POSTGRES_USER' --dbname aiops_topology --command "select *
from types"'
```

The following example output shows that data is present in the inventory (postgres) database:

```
|  id |               type | description |
| --- | ------------------ | ----------- |
|   1 | undergroundStation |             |
|   2 |        railStation |             |
|   5 | interchangeStation |             |
|  31 |      completeGroup |             |
|  32 |                TFL |             |
| 550 |  waiopsApplication |             |
| 552 |             server |             |
| 553 |                 vm |             |
| 554 |           database |             |
```

5. Clear out the topology and inventory databases.

Important: Make a note of the current scaling of the following core services that access the topology and search databases before you configure and run the script:

```
$release-topology-inventory
$release-topology-topology
$release-topology-layout
$release-topology-merge
$release-topology-status
```

The following command provides the current scaling of core services:

```
oc get pods --field-selector=status.phase=Running --no-headers=true --output=custom-
columns="CNAME:.metadata.ownerReferences[0].name" | grep $release-topology | uniq --count
```

Remember that the **`$release`** prefix for the topology services will be specific to your scenario, for example **`aiops-`**.

a. Set the release name and NAMESPACE as required:

```
release=aiops
NAMESPACE=katamari
```

b. Scale down the core services which access the storage.

```
oc scale deploy $release-topology-inventory --replicas=0
oc scale deploy $release-topology-topology --replicas=0
oc scale deploy $release-topology-layout --replicas=0
oc scale deploy $release-topology-merge --replicas=0
oc scale deploy $release-topology-status --replicas=0
```

c. Retrieve the postgres user and password from the Red Hat OpenShift secret

```
export POSTGRES_PASS=$(oc get secret $release-topology-postgres-user -o
jsonpath='{.data.password}' | base64 -d)
export POSTGRES_USER=$(oc get secret $release-topology-postgres-user -o
jsonpath='{.data.username}' | base64 -d)
```

d. Retrieve the cassandra user and password from the Red Hat OpenShift secret

```
export CASSANDRA_PASS=$(oc get secret $release-topology-cassandra-auth-secret -o
jsonpath='{.data.password}' | base64 -d)
export CASSANDRA_USER=$(oc get secret $release-topology-cassandra-auth-secret -o
jsonpath='{.data.username}' | base64 -d)
```

e. Drop the topology janusgraph keyspace.

```
oc exec $release-topology-cassandra-0 -- bash -c "echo 'DROP KEYSPACE janusgraph;exit'
|cqlsh --ssl -u '$CASSANDRA_USER' -p '$CASSANDRA_PASS'"
```

f. Finally, clear out the inventory data

```
POSTGRES_POD=$(oc get clusters.postgresql.k8s.enterprisedb.io --namespace $NAMESPACE -o
jsonpath="{.items[0].status['currentPrimary']}")
oc exec -it $POSTGRES_POD -- /bin/bash -c 'export PGPASSWORD='$POSTGRES_PASS'; psql --
host localhost --username '$POSTGRES_USER' --dbname aiops_topology --command "DO \
\$\$ \
BEGIN \
    EXECUTE (SELECT '\''TRUNCATE TABLE '\'' \
        || string_agg(format('\''%I.%I'\'', schemaname, tablename), '\'', '\'') \
        || '\'' CASCADE'\'' \
    FROM   pg_tables \
    WHERE    schemaname = '\''public'\'' \
    AND NOT tablename = '\''flyway_schema_history'\'' \
    ); \
END \
\$\$;"'
```

6. Post-validate to confirm that the data has been removed from the topology and inventory databases.

   a. Use the following command to verify that the topology database **no longer** has the `janusgraph` keyspace:

```
oc exec $release-topology-cassandra-0 -- bash -c "echo 'SELECT * FROM
system_schema.keyspaces;exit' |cqlsh --ssl -u '$CASSANDRA_USER' -p '$CASSANDRA_PASS'"
```

   The following example output indicates that the janusgraph keyspace is **no longer** present in the topology (Cassandra) database:

```
| keyspace_name | durable_writes | replication
|
| ------------- | -------------- | ----------------------------------------------------
------------------------------- |
|         aiops |           True | {'class':
'org.apache.cassandra.locator.NetworkTopologyStrategy', 'datacenter1': '1'} |
|        x_in_y |           True | {'class':
'org.apache.cassandra.locator.NetworkTopologyStrategy', 'datacenter1': '1'} |
|   system_auth |           True |   {'class':
'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'} |
```

   b. Use the following command to verify that the inventory database contains **no** data.

```
oc exec -it $POSTGRES_POD -- /bin/bash -c 'export PGPASSWORD='$POSTGRES_PASS'; psql --
host localhost --username '$POSTGRES_USER' --dbname aiops_topology --command "select *
from types"'
```

   The following example output shows that **no** data is present in the inventory (postgres) database:

```
| id | type | description |
| -- | ---- | ----------- |

(0 rows)
```

7. Scale up the pods to the same state as before you cleared out the data.

   Note: The number of replicas can be found using the command mentioned in the **Important** note in Step 5.

```
oc scale deploy $release-topology-inventory --replicas=<number_of_replicas>
oc scale deploy $release-topology-topology --replicas=<number_of_replicas>
oc scale deploy $release-topology-layout --replicas=<number_of_replicas>
oc scale deploy $release-topology-merge --replicas=<number_of_replicas>
oc scale deploy $release-topology-status --replicas=<number_of_replicas>
```

Setting the idle session timeout

You can adjust the idle session timeout for IBM Cloud Pak for AIOps in accordance with your security and compliance requirements. If a user leaves their session idle in a web browser for the specified length of time, the user is automatically logged out of the IBM Cloud Pak Automation console.

Before you begin

To complete this task, you must have one of the following roles:

- Red Hat® OpenShift® cluster administrator.
- Red Hat OpenShift project administrator on the project (namespace) where IBM Cloud Pak for AIOps is installed.

About this task

By default, IBM Cloud Pak for AIOps logs users out of the Automation console after 12 hours. You can edit the **product-configmap** to adjust the following session timeout settings:

- The length of time until a user's session expires (TOKEN_EXPIRY_TIME). When the user leaves their session idle for the specified length of time, the user is automatically logged out of the web client. The default is 12 hours.

  If you set **TOKEN_EXPIRY_TIME: "1"**, a user's session will expire after 1 hour of inactivity. If you set **TOKEN_EXPIRY_TIME: "0.5"**, a user's session will expire after 30 minutes of inactivity. You are recommended to set the value between 0.1 and 1.

- The length of time that a user has to refresh their session (TOKEN_REFRESH_PERIOD). The default is 12 hours. You are recommended to set the value between 1 and 24.

  If you set **TOKEN_REFRESH_PERIOD: "1"** and the user's session does not expire, the user's session is automatically refreshed during this 60 minute period. The session is extended based on the value that is set for the **TOKEN_EXPIRY_TIME** parameter. However, after the token refresh period passes, the user must log back into the web client when their current session expires.

  If you don't want to allow users to extend their sessions, set the value of the **TOKEN_REFRESH_PERIOD** parameter to a value less than the value of the **TOKEN_EXPIRY_TIME** parameter.

For example, the following configuration shows potential values for these settings:

```
TOKEN_EXPIRY_TIME: "0.5"
TOKEN_REFRESH_PERIOD: "2"
```

If a user starts work at 8 AM and logs in to the console, the user must be active in the web session within 30 minutes for their token to be refreshed. If the user stops using the Automation console at 8:10 and attempts to use the console again until 8:41, the user must re-authenticate to the console because their session expired.

However, if the user remains active in their session and their token refreshes at 9:59 AM, their session lasts until 10:29 AM. When the session expires at 10:29, the user must re-authenticate to the console because the token refresh period expired.

Procedure

1. Log in to your Red Hat OpenShift Container Platform cluster:

   ```
   oc login OpenShift_URL:port
   ```

2. Change to the project where IBM Cloud Pak for AIOps is deployed:

   ```
   oc project Project
   ```

3. Run the following command to edit the **product-configmap** file:

   ```
   oc edit configmap product-configmap
   ```

4. Add an entry for the **TOKEN_EXPIRY_TIME** parameter to the data section of the **product-configmap** file. For example:

   ```
   data:
     ...
     TOKEN_EXPIRY_TIME: "1"
     ...
   ```

5. Add an entry for the **TOKEN_REFRESH_PERIOD** parameter to the data section of the **product-configmap** file. For example:

```
data:
  ...
  TOKEN_REFRESH_PERIOD: "1"
  ...
```

6. Save your changes to the **product-configmap** file.

7. Restart the **usermgmt** pods for the changes to take effect. To restart the pods, run the following command:

```
oc delete pod -l component=usermgmt
```

Configuring the resources for topology for a production deployment

Ensure that you are logged in to your Red Hat OpenShift cluster with **oc login** for any steps that use the Red Hat OpenShift command-line interface (CLI).

Configuring the File observer

Additional configuration is required if you are using the File observer for more than 600,000 resources.

If IBM Sales representatives and Business Partners did not use the custom sizing tool to supply you with a custom profile ConfigMap to customize your deployment, then run the following command:

```
cat <<EOF | oc apply -f -
apiVersion: v1
kind: ConfigMap
metadata:
  labels:
    app.kubernetes.io/instance: aiops
    app.kubernetes.io/managed-by: ibm-aiops-orchestrator
    app.kubernetes.io/name: custom-sized-profiles
    app.kubernetes.io/part-of: ibm-aiops
    aiops-custom-size-profile-version: 4.2.0-2023
  name: aiops-custom-size-profile
data:
  # WARNING: Modifications to this ConfigMap may cause your AIOPs installation to become
unstable.
  profiles: |
    generatedfor: HA
    configmaps:
    - name: aiops-topology-sizing
      data:
        asm: |
          file-observer:
            specs:
              replicas: 1
            containers:
              file-observer:
                resources:
                  requests:
                    cpu: 100m
                    memory: 750Mi
                  limits:
                    cpu: 1000m
                    memory: 1750Mi
                env:
                  - name: JVM_ARGS
                    value: -Xms300M -Xmx1400M
EOF
```

If IBM Sales representatives and Business Partners used the custom sizing tool to supply you with a custom profile ConfigMap to customize your deployment, then use the following steps to adjust the value of *JVM_ARGS* for the File observer to *-Xms300M -Xmx1400M*:

1. Log in to your Red Hat OpenShift cluster's console, and then from the **Project** list, select the project (namespace) that IBM Cloud Pak for AIOps is deployed in.
2. Navigate to **Workloads** > **ConfigMaps**.
3. Select the name of your custom profile, usually **aiops-custom-size-profile**, and then click **YAML** to switch to the YAML view. Adjust the value of *JVM_ARGS* for the File observer to *-Xms300M -Xmx1400M* and then click **Save**.

   Example excerpt:

```
- name: aiops-topology-sizing
  data:
    asm: |
      file-observer:
        specs:
          replicas: 1
        containers:
          file-observer:
            resources:
              requests:
                cpu: 100m
                memory: 750Mi
              limits:
                cpu: 1000m
                memory: 1750Mi
            env:
              - name: JVM_ARGS
                value: -Xms300M -Xmx1400M
```

Configuring probable cause

The probable cause capability identifies the alert with the greatest probability of being the cause of the alert group, by using a combination of text classification and analysis of the topological information within the alerts.

You can configure how the system classifies alerts and performs probable cause scoring by changing the default values in the API.

Changing default settings of probable cause

Edit MIME Customization APIs in Swagger to change settings.

- Add keywords
- Update keywords
- Set severity
- Update severity
- Other

Next steps

Access Mime Customization APIs with your Platform UI authentication token or API keys. Include the Platform UI access token in the authorization header of your API calls.

For more information on the available API and related Swagger documentation, see APIs.